Data Processing in a Small Transit Company Using an Automatic Passenger Counter

by

Umesh D. Avadhani

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Civil Engineering

APPROVED:

Antoine G. Hobeika, Chairman

John W. Dickey

Charles E. Nunnally

January, 1986

Blacksburg, Virginia

Data Processing in a Small Transit Company

Using an Automatic Passenger Counter

by

Umesh D. Avadhani

Antoine G. Hobeika, Chairman

Civil Engineering

(ABSTRACT)

This thesis describes the work done in the second stage of the implementation of the Automatic Passenger Counter (APC) system at the Roanoke Valley - Metro Transit Company. This second stage deals with the preparation of a few reports and plots that would help the transit managers in efficiently managing the transit system. The reports and plots give an evaluation of the system and service operations by which the decision makers can support their decisions.

For an efficient management of the transit system, data on ridership activity, running times schedule information, and fare revenue is required. From this data it is possible to produce management information reports and summary statistics.

The present data collection program at Roanake Valley-Metro is carried by using checkers and supervisors to collect ridership and schedule adherence information using manual methods. The information needed for efficient management of transit operations is both difficult and expensive to obtain. The new APC system offers the management with a new and powerful tool that will enhance their capability to make better decisions when allocating the service needs. The data from the APC are essential

for the transit property's ongoing planning and scheduling activites. The management could easily quantify the service demands on a route or for the whole system as desired by the user.

ACKNOWLEDGEMENTS

The author would like to dedicate this work to his parents without whose encouragement this would not have been acheived. Mere words are not sufficient to express the support received from the other members of his family.

The author wishes to thank very specially his research advisor, Dr. Antoine G. Hobeika, for his continued guidance and support. The help, encouragement and the guidance received from both, Dr. Charles E. Nunnally and Dr. Antoine G. Hobeika are invaluable. A special note of thanks is forwarded to Dr. John W. Dickey for his guidance in research and course work.

The author also thanks and , of the Roanoke Valley Metro, for the help and cooperation rendered during the installation and testing of the equipment.

The author thanks all his friends who stood by him during his thick and thin days. Thanks is also extended to his graduate fellows in Transportation, and to all the new friendships that were made during this time in Blacksburg. With them the author has spent some of the most beautiful and memorable days.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

1.1 BACKGROUND

The change in planning emphasis from intensive transit improvements to short range transit efficiency actions, plus the growing fiscal pressures have increased the importance of transit system surveillance. A significant amount of information is necessary to properly evaluate a systems' performance and identify potential improvements. Thus the controllers of the transit schemes face the increasingly complex task of determining efficient routes, schedules, etc. To be able to perform such tasks, the transit system management require detailed and accurate information an transit ridership and operations. At the same time it is important to design a data collection program to obtain reliable data at a reasonable cost. In order to do this, the transit managers need answers to questions such as how much data should be collected (i.e. what sample size should be obtained, which data collection techniques are most appropriate, and how often should it be collected (i.e. once a year or at every schedule change).

The information collected is used by the management to design routes and schedules, assess individual routes and systemwide performance and to develop few reports such as the Section 15 Report, Time Point Trip Report, etc.

The traditional approach has been to obtain this informaion using manual checkers to count the passengers and to record the data. Since

transit systems have limited funds, financial support has to be proveded by the Federal Government for collecting and recording data. Transit services are predominantly owned by local governments with federal assistance. But the prevailing economic conditions are forcing the Federal Government to reduce aid for Federal programs. Thus without federal aid, the transit systems have to undergo the data collection program with their own funds.

If the costs for the data collection programs cannot be met with, it would leave the transit managers to guess about the condition of the transit system. If their guess is not right the mistake could make the system pay for a lot of unnecessary and useless information.

So obviously there is a tradeoff. The data collection program is not essential for the functioning of the transit system but the knowledge of the actual service needs can sometimes save more money then that spent on the data collection program. Manually collected data is becoming obsolete. The new trend is to install an automatic data collection program and avoid the problems that are associated with the hiring of personnel for collecting data.

This view is shared by the Federal Government which has published documents and is giving support for the implementation of Automatic Passenger Counters (APC). (6)

1.2 OBJECTIVES OF THE RESEARCH IN THE FIRST STAGE

The city of Roanoke, which is the owner of the local transit system, Valley-Metro, decided to implement an Automatic Passenger Counter system

to collect their ridership data as an experimental mass transit project supported by the Virginia Department of Highways and Transportation. Economic considerations, as well as, faster report generation capabilities of the APC's are the major reasons that made the management study the possibility of acquiring such a system.

The first stage of this experiment dealt mainly with the implementation of the APC system for the Valley-Metro, in which the purpose was to acquire and install an APC system that fits the needs of the transit system under consideration.

In this stage, adequate software was created which would enable the user to decode, edit, manipulate and create the data base from which the reports could be obtained.

1.3 OBJECTIVES OF THE RESEARCH IN THE SECOND AND FINAL STAGE

The main purpose of this stage and this thesis is to create the reports as requested by the Valley-Metro which will reflect some of the particular needs of the system. These reports will be developed from the information obtained in the first stage.

Of all the reports, the most important for the Valley-Metro is the Section 15 Report - Form 406A, which is to be submitted to the Department of Transportation, annually.

To prepare the software to generate these reports is the major task of this thesis. The software would be such that it would enable the user to manipulate and edit the database from the first stage and then create the reports.

This thesis is a documentation of the creation of the software required, which includes

1. A review of the work done in the first stage including the system being followed at the Valley-Metro (Chapter 2);

2. Characteristics of the software developed in the first stage (Chapter 3);

3. A description of the Section 15 and Time Point Trip Reports (Chapter 4);

4. A description of the software developed to obtain the above two reports (Chapter 5);

5. The capabilities of the Lotus-Symphony software package and its utility in this project in generating management reports to support the decisions made by the transit managers (Chapter 5);

6. The different reports and plots obtained and the utility of these in decision analyses (Chapter 6);

7. And finally, a description of the installation and the operation of the APC after its installation, including the procedure to use the programs (Chapter 7).

2.0 LITERATURE REVIEW

2.1 BACKGROUND:

The objective of any transit data collection program is to obtain accurate and reliable information on transit ridership, fare revenue and schedule adherence for individual routes and for the transit system as a whole. Load statistics at each stop is required to identify peak load points and quantify service demands. Both ridership and fare revenue estimation is critical to the budget process. For service provision and scheduling, time checks and running time data are essential. For the design of a new transit service, the maximum load point volumes in the prevailing direction for various time periods during an average weekday is required.

If such information is available then proper evaluations can be made especially if it is accurate and reliable. From this data, potential management and operating benefits, and planning activities could be utilized to minimize the difficulties in the present system and facilitate the decision-makers for effective operation of the system.

Obtaining such information is not an easy task. Since, as mentioned earlier, there are a lot of problems associated with manually collecting this data. There has been an increasing demand for a reliable, cost-effective and faster method for obtaining a system's data. Recognizing the growing need for up-to-date passenger counts and running times checks,

many transit systems are presently investigating the applicability of an APC system.

Since the early 1970's some transit properties have been experimenting with automated data collection equipment.

"The basic function of an Automatic Passenger Counter (APC) system is to acquire accurate data on passenger activity efficiently, as well as the location and time of that activity. This data, which is essential for the transit property's ongoing planning and scheduling activites, must include boarding and alightings of each bus stop, time at that stop, and some measure of the location." (3, p.xiii)

The greatest advantage with an APC system is that data can be made available for practically all the 24 hours of the day. Also the data can be used to prepare reports so that decisions can be made immediately. The flexibility attained using APC systems is superior to manual data collection program because of the automatic recording of information, thus speeding up the preparation of ridership reports, schedule reports, etc.

One main problem with APC's is the non-uniformity that exists among its many applications. The equipment is designed to suit the needs of the particular transit system. In the open market it is difficult to find a system which could be applicable to any system. This has made the APC market unstable and unreliable. Now it is slowly being realized and the designers are in the process of standardizing the equipment's functional requirements. Red Pine Instruments, Canada, is one such company which is standardizing the equipment.

The earliest APC applications used had Automatic Vehicle Monitoring (AVM) systems. AVM systems used infrared beam counters to record data on passenger boardings and alightings. This system is designed to provide

continuous information on vehicle location and schedule adherence. Information is constantly transmitted by radio communications to a central processing center. This information is then used for service monitoring and performance evaluation. Some AVM's has the capability to store information for later retrieval, validation and processing.

After the initial experiments, transit properties began implementing off-line APC systems, to obtain the ridership and schedule activities at each bus stop. Basically, the collection of data is similar, but it is stored in some type of memory device; to be processed at a later time.

2.2 APC'S IN GENERAL AND AT VALLEY-METRO IN PARTICULAR

The basic function of an APC is to record and register passenger activity, the time of arrival at the stop and some point of location reference. The basic steps are as shown in Figure 1 . In collecting data with an APC system it should include the number of boardings, alightings, the time at each bus stop and a distance measuring device.

1. Data Acquisition

Acquisition means sensing and recording data. The passengers getting on and off are detected at each doorway. Such a significant count must be timed and recorded. At this time the distance from odometer must be recorded.

At Valley-Metro, passengers are sensed as the photoelectric beams are cut off. For example, if the outside beam is activated first, after all the four beams were cut off, it is recorded as an "On", and the vice-versa

holds good for an "Off". There is an internal clock which stamps the time and the distance is recorded from the pulses emitted from the wheel.

2. Data Recording and Storage

The data generally recorded and stored by an APC system includes:

- passengers boarding and alighting at each stop;
- the time associated with the arrival at each stop where passenger activity occurs;
- the distance associated with the arrival;
- location referencing points which might include
  - a signpost-transmitted signal
  - a switch to create a record at the end of every trip or loop

At Valley-Metro, an on-board microprocessor is electronically connected to interpret the signals from the sensors in terms of passengers alighting and boarding, depending on the direction of the beam being cut off. These records are stored in the memory, to be transfered to a 3.5" x 3.5" diskette.

3. Data Transfer

There are different ways of transferring data from the microprocessor, by radio transmission through cassette tapes or by means of a portable microcomputer.

At Valley-Metro a portable data general microcomputer is used for data transferring. A 3.5" diskette is used for storage as a temporary storage medium, before it is transferred to a floppy disk and then to a computer.

4. Data Processing, Analysis and Reporting

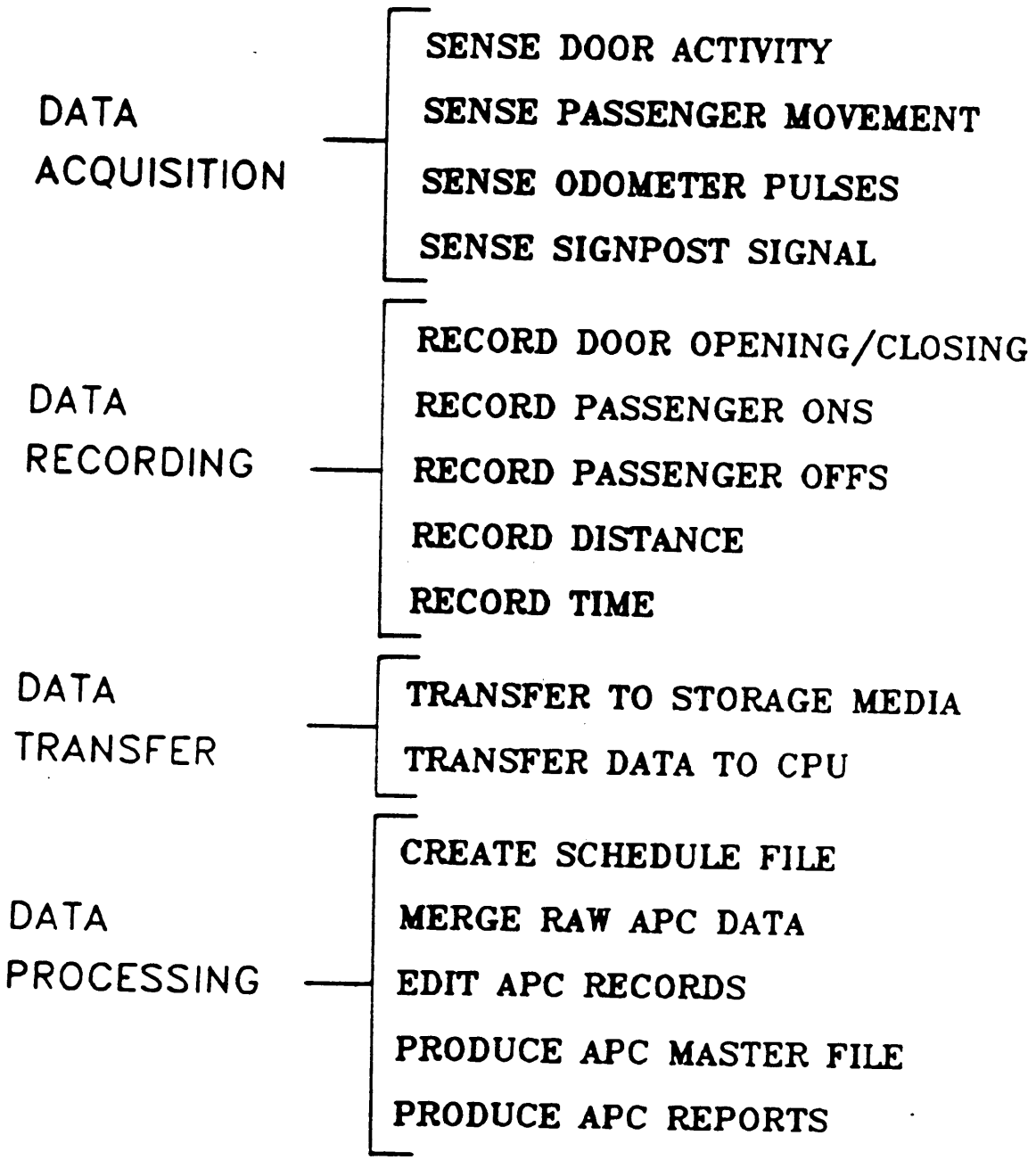

Figure 1. APC's Basic Steps
(Ref 3, p. 34)

The data is processed and analyzed using suitable software to generate the desired reports. The first step, usually, is related to referencing the vehicle trip and stop for each of the APC records.

Referencing is easier at Valley-Metro since there is a signpost record (at present it is still not installed). Now there is a manual input record which is created by the operator at the end of a trip or loop. A master file is created with stop by stop statistics using a few external files.

2.3 EQUIPMENT IN GENERAL, AND AT VALLEY-METRO IN PARTICULAR

Considering the various components involved in an APC system, the equipment could be divided into two major components

1. Data Gathering Equipment
2. Data Processing/Analysis Equipment

By 'Data Gathering Equipment' is meant all the hardware on the bus. The Data Processing/Analysis Equipment is the software to be used in the computer to generate ridership and schedule reports.

2.3.1 Data Gathering Equipment

At Valley-Metro the following equipment are installed:

1. Essential Modules

a. System Controller

b. Memory

c. Passenger Counters

2. User-Selected Modules

a. Manual Input

b. Status Display

c. External Data Retriever

2.3.1.1 Essential Modules

The following is a description of the essential modules, mentioned above, installed at Valley-Metro. Each equipment was later modified to suit the needs of the Valley-Metro system.

a. System Controller: This is the "brain" of the whole system.

"It is a small microprocessor that accepts, monitors, and controls the data collection and data transfer functions of all other modules. It also includes a clock. It accepts data directly from the odometer and the driver door control switches." (2, p. 21)

The module, installed at Valley-Metro, is a Motorola 6803P microprocessor developed by Red Pine, which is capable of interpreting, recording and transferring the data to a memory device. It has incorporated an electronic timer that records the time each time a particular event occurs. This microprocessor uses a logic algorithm programmed inside its memory. With the help of this algorithm, the APC system will translate this information into a number of boardings and alightings. At this moment, time is recorded along with an odometer reading for each stop where passenger activity occurred.

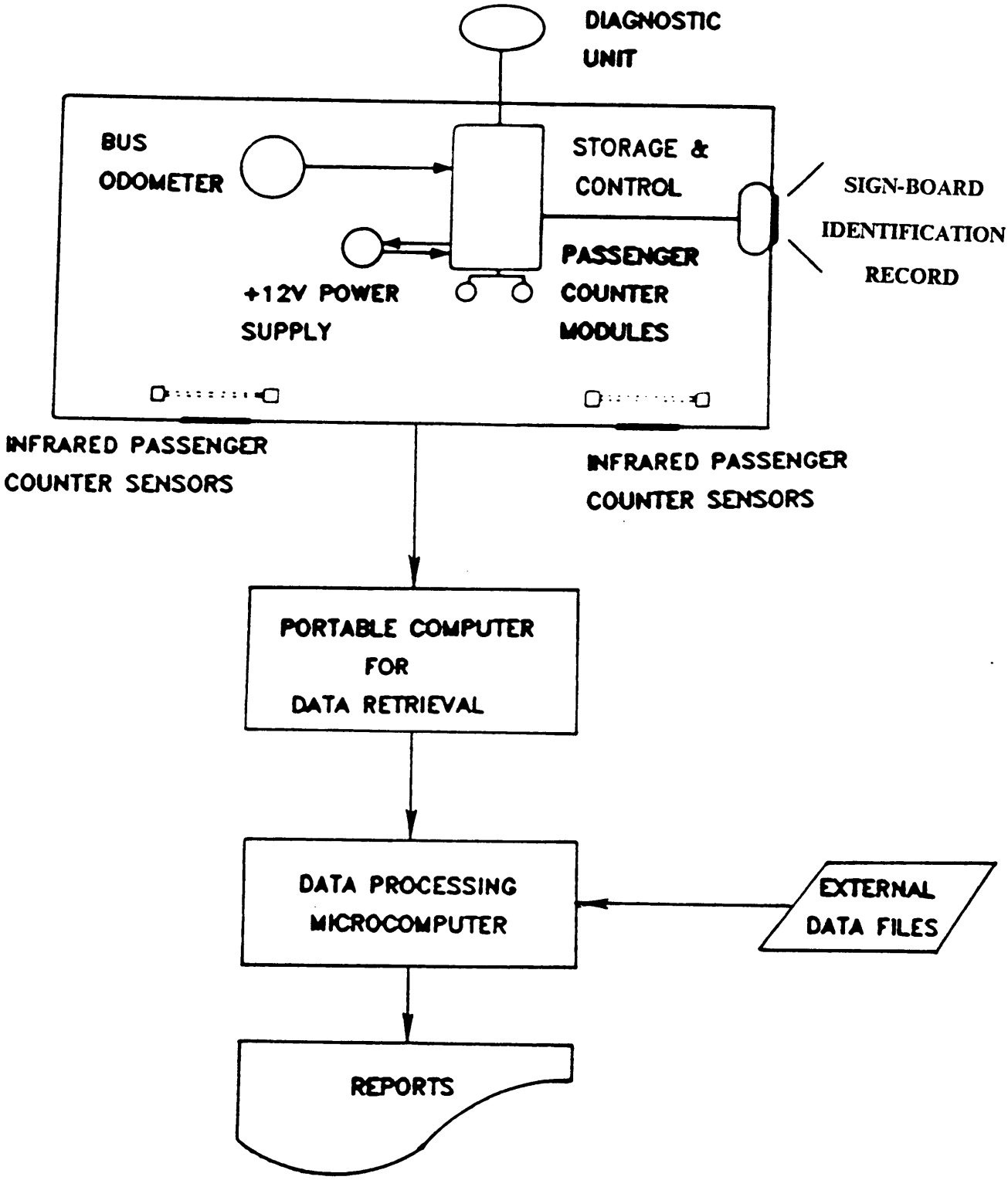

Figure 2. Valley-Metro System Components

The logic algorithm automatically instructs the microprocessor to record the different data as and when a particular event occurs. The "logs" that are recorded is shown in the Figures 3 and 4.

In the first stage it was decided to install a signpost transceiver and transponder. But due to some manufacturing problems, it has not yet been supplied. Hence it was decided to create a similar log at the end of a trip. This "log" is created whenever the operator of the bus changes the name of the route. This record, thus, gives a reference point such that the odometer transitions could be referenced with the external files. The memory of the microprocessor is a solid state memory to prevent any data loss that may occur due to mishandlings.

Logs are fixed length (e.g., 5 bytes) records of data. They are generated whenever any of the following events occur

- Begin of idle
- End of idle
- The beam of the sensors are interpreted
- The route sign is changed
- An odometer impulse

Thus the data records are stored in the memory in a "log" form consisting mainly of:

- elapsed time (in multiples of 15 seconds)
- distance travelled (in odometer impulses of about 11.6 ft)
- total passengers boarding

- total passengers alighting
- the reason the log was taken.

The distance is recorded in units of 11.6 ft since an impulse is given out by the odometer every 11.6 ft. Actually this can be any other value since it could be adjusted in the circuit of the hardware.
Logs will be recorded for five reasons:

- Log 2: the space in the memory allocated to record time is about to overflow and must be reset (this occurs everyhour)
- Log 4: the space in memory allocated to record distance is about to overflow and must be reset
- Log 3: the bus has been "idle" for one minute, i.e., has moved less than 50 feet without passengers either boarding or alighting.
- Log 6: the bus moves 50 feet after having been idle for two minutes or more
- Log 5: at least one passenger boards or at least one passenger alights and the bus moves at least 50 feet
- Log 11: a manual input log was detected whenever the operator of the bus changes the signboard to indicate the change in the route.

All logs are 5 bytes long. The first byte gives the type of log. Whenever time is recorded, the second byte is used and represents the elapsed time since the previous log. Distance when used is the third byte and represents the number of odometer transitions since the previous log.

Passenger counts represent the number of passengers counted since the previous log.

b. Passenger Counter Module/Sensors:

There are different types of passenger counting sensors available in the market.

1. Switch Mat Sensors

These are pressure sensitive mats installed on two steps of each door. Depending on which mat is stepped first, determines the direction of movement hence giving an on or an off.

2. Inductive Loop Mats Sensors

These mats are similar to the switch mats, but instead of pressure activating the switches, they detect passengers with a non-contacting inductive proximity switches.

3. Photoelectric Beam Sensors

These are infrared light beams transmitted through a transmitter to be received by a receiver. These sensors are located at each doorway to detect the boardings and alightings. Each door needs at least two pairs of sensors.

At Valley-Metro, these type of sensors are installed. This module is capable of decoding the pulses from the infrared beams and convert it into passenger ons or offs, as the case may be. An on or off is recorded, depending on the sequence the beams are broken.

There are chances of handbags, umbrellas or hands of the passenger being counted as passengers. This is prevented by activating the sensors only when the doors are open. The sensors on the same side are skewed 6" horizontally and vertically, so as to not to miss a thin person. The

1	POWER ON / RESET
2	HOUR OVERFLOW
3	VEHICLE STOPPED FOR ONE MINUTE
4	DISTANCE OVERFLOW (256 x ODOMETER TRANSITION)
5	GENERATED BY THE FIRST ODOMETER TRANSITION AFTER PASSENGERS ARE COUNTED
6	GENERATED BY FIRST ODOMETER TRANSITION AFTER VEHICLE HAS STOPPED FOR GREATER THAN 2 MINUTES
7	THE ELAPSED TIME IN SECONDS SINCE THE PREVIOUS DUMP / A POWER ON / RESET
8	THE DISTANCE IN ODOMETER TRANSITIONS SINCE THE PREVIOUS DUMP/ POWER ON / RESET
12	MANUALLY INPUT LOG MONTH-DAY-YEAR-ROUTE ID
12	MANUALLY INPUT LOG DAY OF THE WK-PERSON ID-HOURS-MINUTES

Figure 3. Functions of the Logs

REASON	TYPE	TIME	DIST	ONS	OFFS
POWER ON/RESET	1	0	0	0	0
HOUR OVERFLOW	2	0	0	0	0
ONE MINUTE IDLE	3	X	X	0	0
DIST. OVERFLOW	4	X	X	0	0
PASSENGER LOG	5	X	X	X	X
END LONG IDLE	6	X	X	0	0
FORCED LAST TIME	7	X	X	X	X
FORCED LAST DIST	8	0	X	X	X
SIGN-BOARD RECORD	11	X	X	0	0
MANUALLY INPUT	12	MO	DAY	YR	BUS ID
MANUALLY INPUT	12	WKDY	PRS.ID	HRS	MIN

Figure 4. Types of Logs Recorded

most important thing to be noted is that a passenger count is recorded only if all the four beams are interrupted at the same time. This information is accumulated in the module until accessed via the module interface to the system controller.

2.3.2 Stationery Equipment (User-Selected Modules)

a. Manual Input: This log is inserted by the user whenever one dumps the data. Information such as time, date, day of the week, run number, route number, etc., could be fed.

Also the log created when the route direction is changed is also a sort of manual input log. This log identifies the end or beginning of a trip.

b. Status display: This is a diagnostic unit which is a portable unit. This displays the pattern in which the beams are broken, the time, the distance and the passengers getting on or off.

This is performed on the bus when the APC is in operation. Any error in the system could be identified by this unit.

c. External Data Retriever: This unit is to retrieve the data or records from the memory. At Valley-Metro to perform this function, a Data General microcomputer is used. This microcomputer is directly plugged in the memory and the retrieved data is stored on a 3.5 inches diskette. This data is then used in the office on a floppy disk and a personal microcomputer, to generate the required reports using the adequate software developed. Whenever the data is dumped from the memory to the

diskette, the manual input log is requested. The required information is keyed in to end the retrieval process.

2.4 CURRENT PROCEDURES FOR PASSENGER COUNTS AT VALLEY-METRO

The Roanoke Valley-Metro Transit System receives federal funds to carry out a data collection program, that samples every trip in a typical weekday and a Saturday. This data collection program is carried out by an outside agency. The data is manually collected by about 8-10 checkers who ride on the bus. This agency takes about 2-3 weeks to collect the required data, and it is carried out only once a year, usually in the Fall.

The checkers carry forms on the buses and record the passengers boarding or alighting at each stop. At the same, the arrival time and the odometer reading are also recorded. To make schedule adherence checks, the arrival time is compared to the scheduled arrival time at several time points along each route. The filled up forms are then sent to the office at Valley-Metro, wherein all the counts are corrected for any errors that were made in the data collection. These are then aggregated to create the reports for each bus trip sampled to represent a whole day's data of the bus run along a route. This whole process from the data collection to the time when the final reports reach the transit managers at Valley-Metro, usually takes no less than four months.

Many transit agencies have reported that they faced some problems with this system of data collection program. During peak hours, due to heavy

loading, it becomes difficult to count the exact ons and offs. The checkers, thus, miss a few people hence recording wrong counts.

2.4.1 Data Reliability

Manual data collection is very cumbersome and uneasy. Since ridership activities change according to the season, it is not possible to collect data for each season because of the system's limited resources. This type of data collection does not represent data for the whole year. It only represents information for a couple of days per route, only. Also, transit ridership changes every day and varies according to weather conditions, day of the week, holiday, school day, etc. Therefore, the services cannot be estimated for their adequacy depending only on this data.

Validation of the data collected is not carried out in any way. If any errors are present, a rough approximation is made thus making the data inaccurate. The final load should be zero, in a day. To achieve this, approximations are made thus giving inappropriate stop statistics.

2.4.2 Costs Involved

The exact cost for this type of data collection program is very difficult to estimate. Only a rough estimation of the costs for the program was obtained from the 5th District Planning Commission. The cost of the personnels involved to aggregate, analyze and edit the final reports was not available at Valley-Metro. An approximate cost of $10,000 was given

by the agency for the checkers who ride the bus. This information, about costs, is too rough an estimate to make any kind of evaluation.

3.0 CHARACTERISTICS OF SOFTWARE IN THE FIRST STAGE

The software was designed to create a database from the information obtained from the APC. During the first stage the equipment had not been delivered, but the manufacturer lent a similar equipment so that suitable tests and experiments could be carried out. With this equipment and an electronic odometer simulator, it was possible for the research team to become acquainted with most of the functions of the equipment.

3.1 TESTS AND EXPERIMENTS

The project was started around November, 1984, and it was decided to carry out experiments using the borrowed equipment till the new one arrived.

Red Pine Instruments provided a microprocessor with passenger counter modules and sensors which records data just like the new one. It was thus possible to create the initial software needed to edit raw data files and create a suitable database from which the final reports can be produced. To obtain the database, a few external data have to be referenced.

3.2 DESCRIPTION OF THE EXPERIMENTS CONDUCTED

Data was obtained from the manually collected data and it was edited in such a manner as though it was actually obtained from the APC. This was possible because of the borrowed equipment.

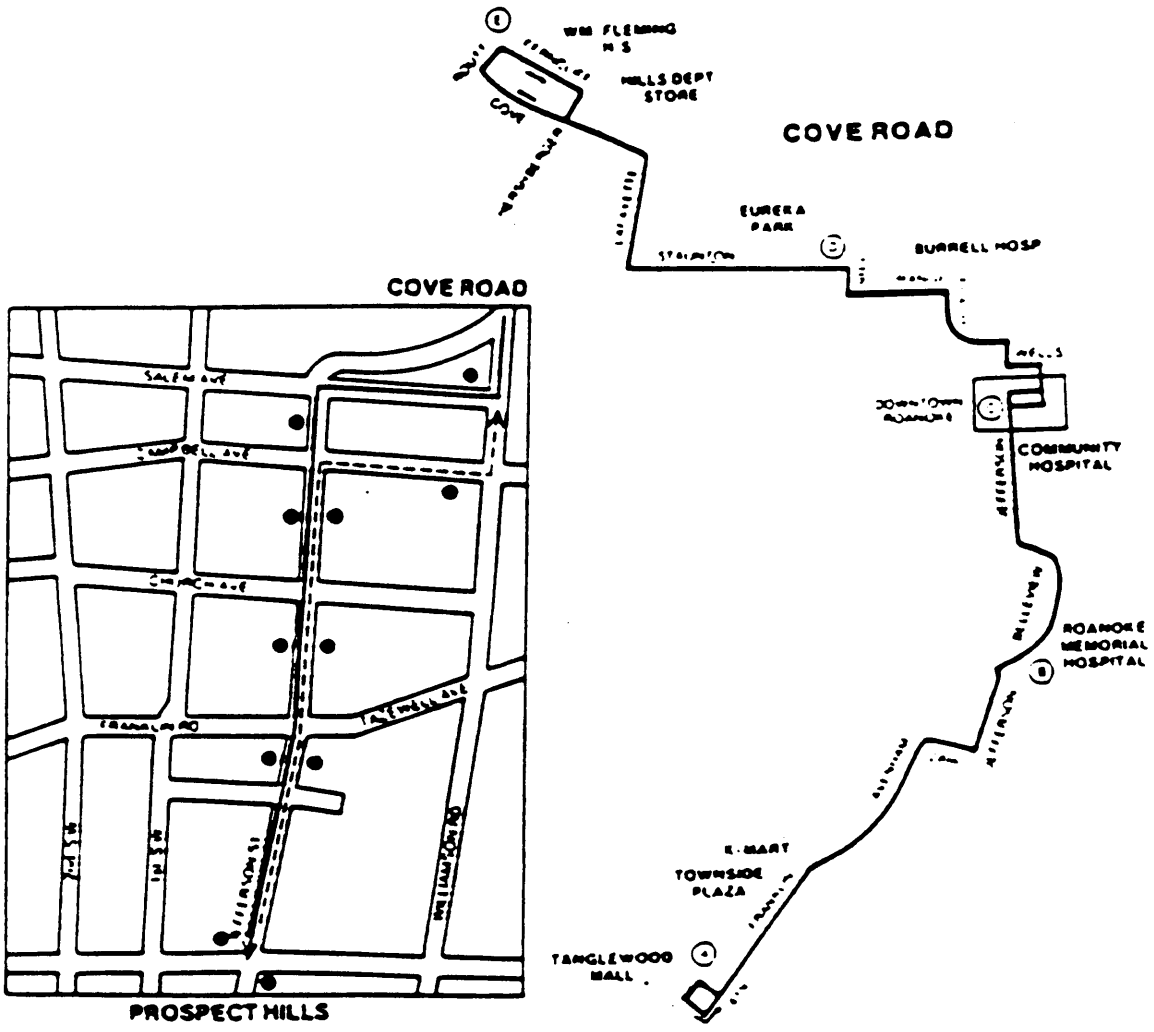

Figure 5. Cove Road/Prospect Hills Bus Route

This manually collected data was obtained from the sheets at Valley-Metro. On these sheets there were the ons and offs at each stop. This data was then manipulated and the time and distance were measured so that there was an idea as to the actual distances traversed by the bus.

To get the scheduled times at some of the stops along the route, the schedule sheet at Valley-Metro was referred to. This schedule data file is is the one which contains the scheduled times for a few check points along the route. Scheduled times at Valley-Metro are obtained for a particular block number of the day.

The data obtained from the sheets was for the Core Road/Prospect Hills route. Actually there are 10 routes operating daily at Valley-Metro. It was on the Core Road/Prospect Hills route, that the Valley-Metro managers decided to test the APC equipment. A complete round trip is about 20.7 miles and has 209 official stops.

Distance matching has to be done to get the stop statistic whenever the bus stopped. For this matching, the exact distance in feet between each stop should be known.

With the help of the Virginia Highway and Transportation Department, which has some special vehicles that are equipped with a special odometer than can measure distances up to a foot, it was possible to know the distance between each stop. During this stage, it was decided to install a signpost, but due to some manufacturing delays it is not installed as yet. Instead of a signpost, there is a similar log created whenever the bus reaches the end of a trip. This log is created when the operator changes the signboard to indicate the direction of the route. With this type of record we know exactly the location of the bus and the distances

are then referenced since the last trip change log. This always provides a better matching process. In fact if there are signposts the matching process is reliable and usually accurate.

On this route, all the stops do not have a name. Hence it was decided to name the stops after the intersecting streets to which the stops were closer. The same principle is followed for the data collected manually.

To match the data with the distances, an external file of distances with the stop names was created, numbering each of the stops. When the data referencing is made during the execution of the software program this file is referenced to, for further processing.

3.3 SOFTWARE DESIGNED IN THE FIRST STAGE

Most of the reports that are requested by the Valley-Metro managers are similar to the ones made in Cincinnati by the Urban Transportation Associates. Thus, the reports are modelled after the reports made by the UTA.

3.3.1 General Data Flow at Valley-Metro

In the first stage it was decided that the automated data collection program should be at least similar to the actual program. The software should be so designed as to create reports as Valley-Metro gets them every year.

This stage dealt mainly with the data processing requirements such that it will provide the primary archives of raw reports from which the

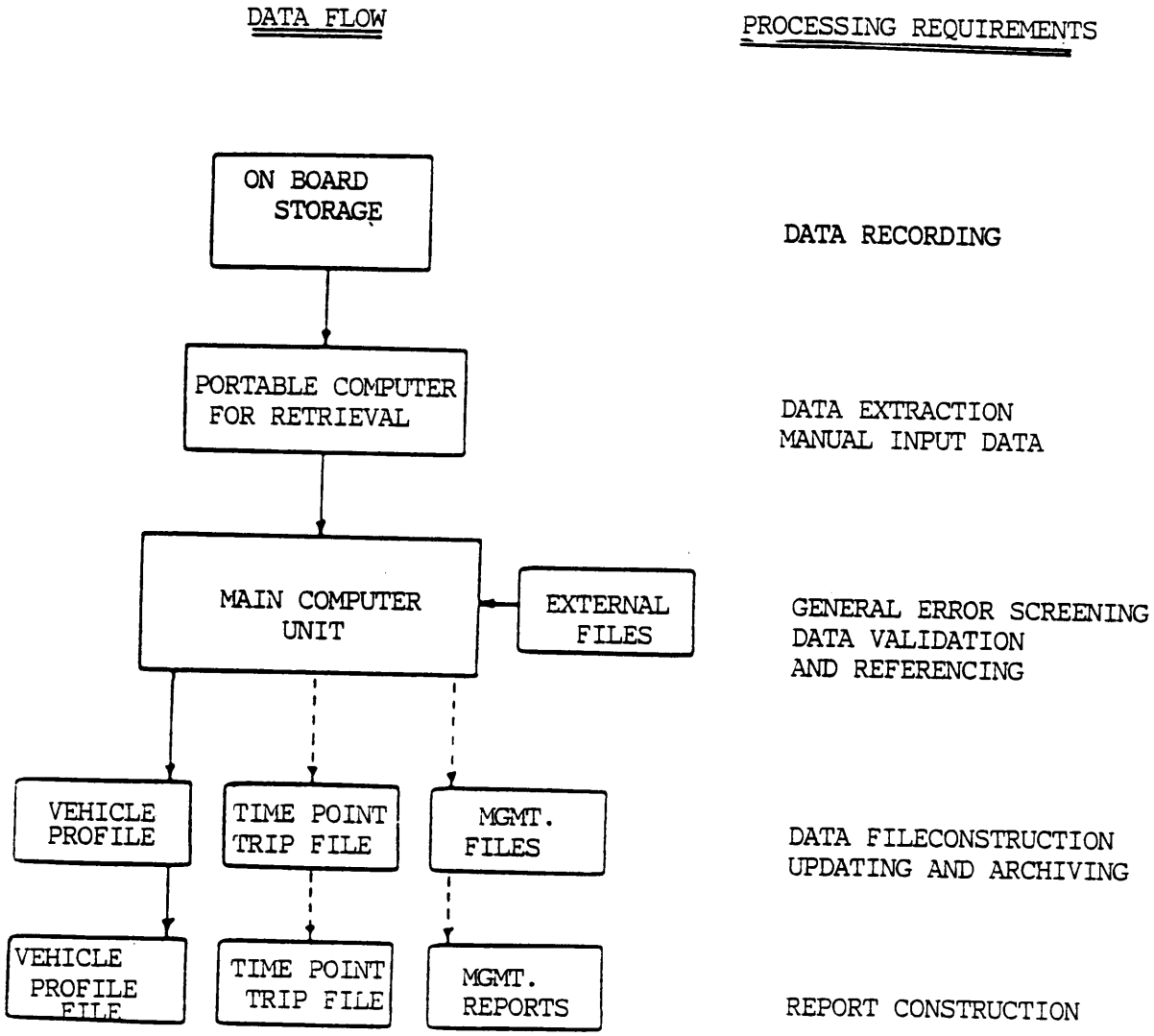

Figure 6. General Data Flow at Valley-Metro

required management reports could be obtained. Figure 6 shows the general data flow for this project. The second stage involved the creation of reports that will be used at the higher level of management decision making.

The data obtained from the APC is then processed in the Valley-Metro office. For this processing, certain external files such as schedule file and distance file, which contains distances between stops, have to be referenced to validate the data and reference it to actual bus stop locations.

The file, Vehicle Profile File, contains the primary sources of raw data. This file contains the data associated with a bus assignment in a particular day, including the statistics at each stop. The stop statistics include the arrival time, number of ons and offs, the load and distance travelled, at each stop for a particular day.

3.4 COMPUTER PROGRAM DESCRIPTION

The following section describes in detail the computer programs that were developed to achieve the data processing that create the vehicle profile file and some of the reports that can be obtained from this file.

3.4.1 Decoding and Data Manipulation

As it was explained before, the data logs obtained from the on-board memory will be 5 byte records. The first byte is a number which identifies the type of log it is, and bytes 2 to 5 have information that depends on

the type of log it is (see figure 3). To explain better how the program works see figure 7, which is a general flowchart of the program.

On the left side is the flowchart, and on the right side is the name of the subroutine associated with the each data processing task. The names of the subroutines were chosen to represent the type of processing activity they perform in the program. This way, the program becomes more self-explanatory. A copy of the program is listed in appendix A.

The program starts with the MAIN program which reads all the data from the input file. There is a preliminary screening for flagrant errors in the data. These errors are usually determined by the maximum value that are allowed in each byte, and others defined by the user. Among these, for example, is the maximum number that can appear in byte 5, associated only with the number of passengers getting off. For practical purposes, this value should not be greater than the capacity of the bus, i.e. 60 passengers. This value will then be the upper limit when reading the data. It can be changed to accommodate any size of bus by simply changing the variable's value that checks this limit. The exception is for the last three logs made, which identify the dump request record and the manually inputted data. These limits are changed to check their highest values.

Should there be any error identified in this first screening, there is the subroutine ERRORS which allows the program to change the values on these records where an error was identified. It is optional to make any changes in any of these records.

Processing continues and the user will be prompted for a header record in case the manually input data was not inputted when the data was dumped. The TIMEIN subroutine is the one in charge of this operation.

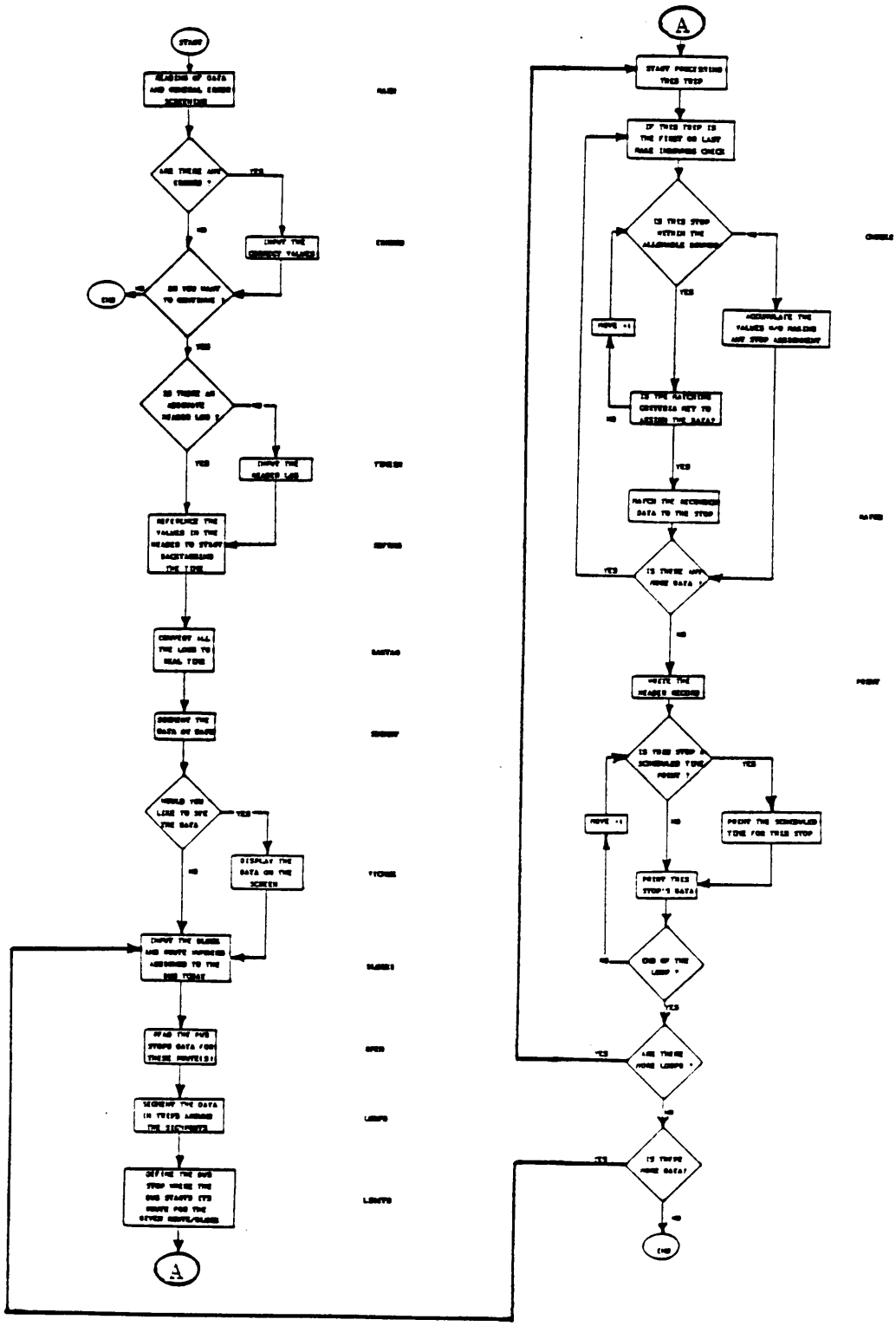

Figure 7. APC Program Flowchart

(Ref. 16)

The program continues with the REFRNS subroutine. Here, the real time and date is defined as the reference time and date from which the next subroutine, BAKTAG, starts backtagging all of the logs from the last to the first, in decreasing order. The result is a real time and date associated with each log in the data collected. This allows the next subroutine, SEGMNT, to segment or divide the data logs into different days.

An optional visual checking of the data is available with the next subroutine, VICHEK. This subroutine is used primarily to verify any changes that could have been made with the ERRORS subroutine. In larger systems, this subroutine is used to identify any splitting of a late bus working after midnight. This data should not be allocated to the next days data. At Valley-Metro all buses are in before 7:00 p.m.

All of the processing from here on is executed for each day, as long as there is data available. This means that the program can handle as many days as there is data. The main factor that will limit the use of this will be the memory in the bus equipment.

Next, in BLOCK1, the user is prompted to input the vehicle assignments for the day. The subroutine asks information on the route and block number of the data that is to be associated with that day. In larger APC systems, this task is done from an assignments file which contains all of the equipped buses assignments in the fleet. This approach was deemed unnecessary for this project because it would mean an extra file that would need to be updated periodically. Valley-Metro is starting with only one equipped bus. It is recommended to have only 10% of the fleet equipped. This means about 3 equipped buses. Having to update another file for 3 buses is really not very practical, because it seems much more easier that

the person in charge of dumping and/or processing the data, look up this information from the dispatcher's sheet every morning, and not typing it in a separate file.

The OPEN subroutine then opens the files from which the distance data and name tag for each bus stop on the routes and blocks specified above. Should there be a change in route/block in one day, the user is prompted for the time in which this change happened and the change of route/block.

The next step is to divide the data into trips. The LOOPS subroutine identifies the starting location of a trip by dividing the days data with the signboard records that are made once during each trip just before the first stop, which is the Brendles bus stop at Tanglewood Mall. From this division, it is known exactly the location in the data when the bus passed through this stop.

Basically, all of the subroutines can be used on data collected for any route, or with more signboards, except for LIMITS and CHEKPT. These two next subroutines are really data files that identify the beginning and end of a particular bus block for a particular route. They determine where the matching of the bus stops data should begin, primarily in the first and the last trip of a bus in one day. The problem is that since not all the bus blocks start from the same location at the same time, some will have to match data before the signboard and others at the signboard. For buses starting at the Brendles stop, there is no problem, since the matching is done forward. But for the buses starting at other locations, there is a problem, because the referencing will start at the signboard record and then start backward to the first log. When this first part is done, the process is the same for all blocks, since the signboard is

changed after one total loop. The problem arises again at the last trip, when some of the buses finish their assignments at places that is not the signboard location.

This problem is solved with the subroutine LIMITS, which defines the range of data logs between which a trip lies. In the first trip it also defines that the matching be done from the last stop to the first, if the first stop for that block does not equal the stop at the signboard location.

The next subroutine, CHEKPT, tries to check if the matching process that is being carried in the program is valid, or in other words, if it is being done within valid limits of the route for the determined block assigned to that bus on that day. This is done especially for the first and the last trip every day, where the first and last stop of the bus will usually not be the first stop of the route (Brendles-Tang. Mall). If this was not done, some of the data could be allocated at wrong locations that do not represent events on the route. A typical example would be, for instance, if a bus driver is required to take some personnel from the bus garage to a certain place **before** the begining of the block's first stop. The program would probably try to reference this event with any stop that is located at the same distance inside the route loop from the first stop in that block's itinerary.

Instead, all of the information that is outside the limits of the first trip, is accumulated and specified in the first record of the vehicle profile file for that day. When this happens in the last trip, the information is accumulated and recorded in the signboard record in the

last trip. The formatting of the vehicle profile record is explained in more detail in the next section.

After this is done for the first data record of the first trip, the MATCH subroutine is called to match the first bus stop record with the first bus stop location that was specified for that block by the LIMITS subroutine.

If the distance measured by the bus odometer is not equal to the distance between the signboard and the first stop, this distance is increased by the distance from the first to the second stop. The program will call CHEKPT in the first and last trip at this stage to verify that this next stop is inside the allowed limits. If it is not, it will accumulate the data for the header/signboard record, as appropriate.

The process of increasing actual distances between bus stops to try to match the measured distance by the bus odometer terminates when either one of the following conditions is met:

1. The difference between the measured distance and the accumulation of the actual distances is less than or equal to one odometer impulse, which we defined as being 11.6 ft. This variable called CLICK in the subroutine MATCH, can be changed when the real value is known.
2. The difference between the measured distance and the accumulation of the actual distances is greater than one odometer impulse, but adding the distance between this stop and the next gives a greater absolute difference in value.

When a match is done, the data on arrival time to the stop is obtained from the door opening log. The departure time and the passenger activity is obtained from the following passenger log. The distance measured and the idle time between this and the previous bus stop that was successfully matched, is also identified to the actual bus stop. The distance matching process will begin for the next passenger record using the stop used in the previous passenger record, as the beginning point for the distance accumulation process.

This sequence continues until all of the data logs for that trip are processed. The next step is to create the vehicle profile file that will mainly be a big matrix, with one header record for each day, and a series of data segments of the matrix, each segment representing one whole trip of the route. The contents of this matrix will be discussed in more detail in the following section. This file will be written with all of the data for the trip by the PRINT subroutine.

When the present loop of the route is finished, the last record in the matrix should be the signboard record, where there will be information that needs to be accumulated for the next trip, such as distance traveled and load. After this step, the subroutine returns to identify the location of the next trips data using the LIMITS subroutine. If there are no more trips, the processing is transferred to the subroutine BLOCK1, where the user will be prompted for the following day's block and route assignments of the data. If there is no more data for another day, the program is terminated.

3.4.2 Editing and Formatting

The output of the first stage is mainly the vehicle profile file, whose use was described before. This file is mainly a big matrix, which consists of a header record and several smaller matrices which contain the information of a trip's data. If the bus should change route and/or block during the day, a new header record is created.

The header record mainly tries to identify the bus route and block identification number, the date, the bus number and the trip number, which is usually 1. This information basically will not change until the next header record. It also has information on the accumulated distance, load, and idle time the bus has until it is officially in service at the first scheduled bus stop.

The following part of the matrix is really a whole trip around a route. It has a sequential number identifying the route stop, and it details all of the information pertinent to that specific bus stop. The data includes the arrival time, the departure time, the number of passengers that got on and off at that stop, and the load of the bus on that stop, the distance the bus has traveled up to that point in the day (or since the change of route/block), a sign, (*), which identifies a successful matching of the distances measured by the bus odometer and the actual distances between stops. If this sign is not present, it means the second alternative in the matching was taken, and that the matching was done, but it did not meet the matching criteria for a "good" match. The next column has any idle time that can be associated between the present bus stop and the last

stop the bus made. And finally, only on those stops where there is a schedule time available, the scheduled time for the bus trip at that stop.

This file will be accessed by future software to create the vehicle specific reports associated with the data sampled on that day. A sample of a report that can be aggregated from this file and the distance data file that has the names of the stops, is included in appendix.

In the second phase of the project, these files will be used primarily as raw data archives. Meanwhile, the sample report is very similar to that obtained from the manually collected data, with the difference that this will be made in 15 minutes after the APC data is fed into the computer.

4.0 DESCRIPTION OF THE TWO REPORTS

This chapter and the latter chapter deals with the second stage of the project. As mentioned earlier, the most important report for the Valley-Metro is the Section 15 report, which has to be submitted to the UMTA, annually by every transit company. This file maintains the passenger ridership information. The Time Point Trip File maintains records for each trip, with stop statistics between time points.

4.1 SECTION 15 REPORT

This report is a mandatory report for all transit system. This report is required for receiving financial aid from UMTA.

Four different report levels have been established, as shown in Fig. 8.

The level of reporting depends on the size classification of the transit company. For example, a transit company with 200 revenue vehicles may report at Level A or C, instead of Level B. As far as the Federal government is concerned, any one of the levels satisfies the reporting requirements unless a transit agency applied for more grants. It is thus obligatory on the reporters to check with their state and/or local authorities before selecting a reporting level.

The Valley-Metro, operates about 30 buses in the morning and evening peak-periods. Valley-Metro reports the Section 15 report on Form 406A. This Form 406 is titled as follows: "Transit System Service Supplied, Service Consumed, Service Personnel, and Operated Schedule (Non-Rail

Reporting Level			
Level R	Level C	Level B	Level A
required level	voluntary level	voluntary level	voluntary level
all systems	suggested for <100 revenue vehicles	suggested for 100-500 revenue vehicles	suggested for > 500 revenue vehicles, and all rapid rail systems

Figure 8. Reporting Levels

(Source: Ref. 13)

Modes)". Form 406 is used to report several types of data relating to the provision of service for non-rail modes. This report contains a measure of the quantity of service supplied by reporters; passenger trips and passenger miles; the number and characteristics of employees used to provide transit service; and the number of days that the system operated during the year.

At Valley-Metro, Form 406A is used for reporting, which is titled simply as -- Annual Report to UMTA. A sample form is shown in Figure 9.

The purpose of Form 406A is to collect operating data for fixed route bus systems. This annual report maintains details such as passengers boarded, unlinked passenger trips, passenger miles, etc. These variable items are obtained for different time periods of a weekday, such as AM peak, Midday, PM peak and other.

4.1.1 Details of the Report

1. Average Weekday:

The term "Average Weekday" is interpreted as being a typical or representative weekday in the operation of the transit system. Using suitable sample methods, an average weekday should be computed as the average of several weekdays selected at random throughout the year. Average Saturdays and Sundays should be computed the same way.

2. Time Periods:

- AM Peak
- Midday

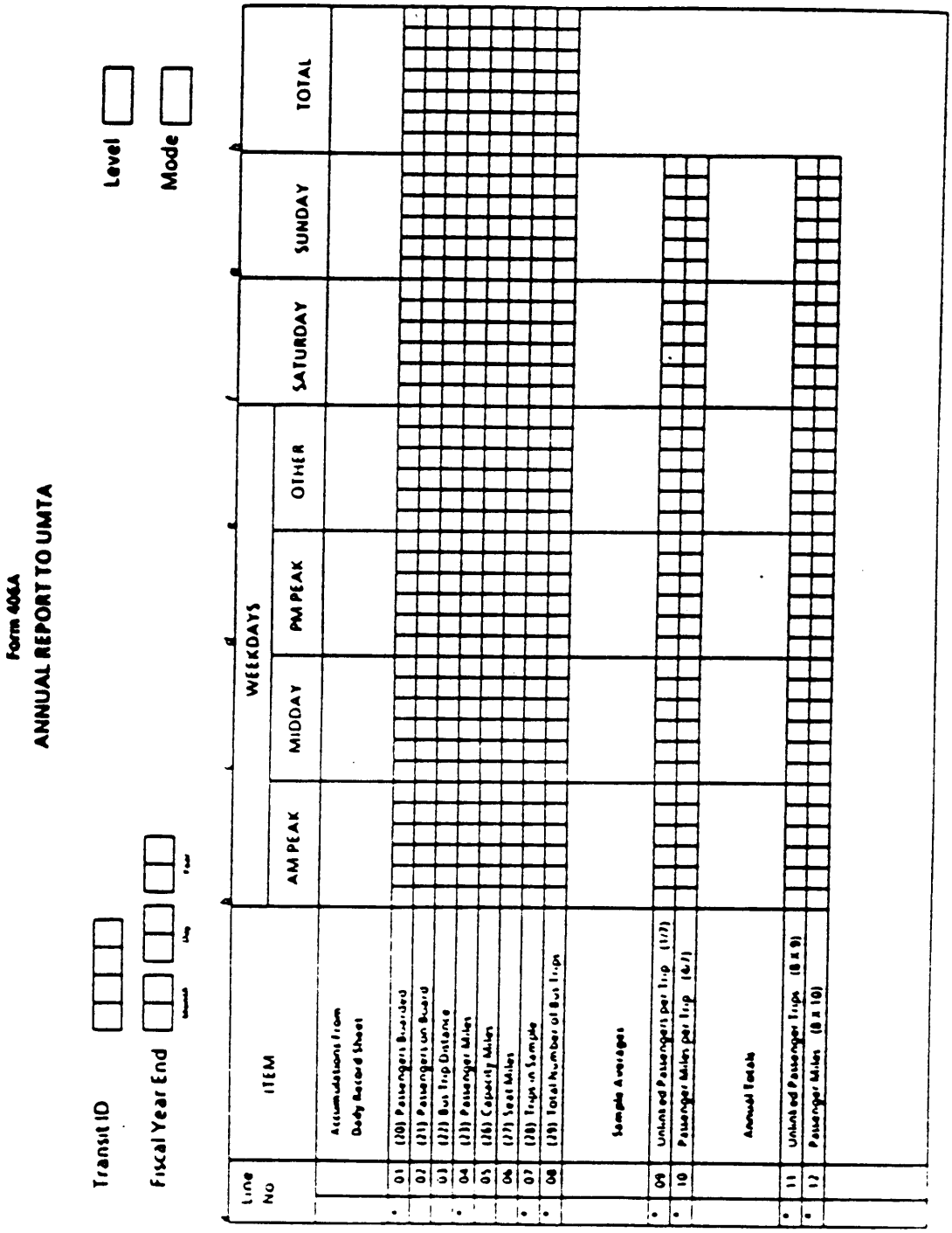

Figure 9. Section 15 Reporting Form

(Ref. 9)

- PM Peak
- Other

AM Peak means the morning peak period. At Valley-Metro, the AM peak starts from 6 am to 9 am. Midday is the period between 9 am and 3 pm whereas PM peak is the period between 3 pm to 7 pm. "Other" combines any service before 6 am and after 7 pm. Usually all the buses at Valley-Metro are back into the barn by 7 pm.

3. Annual Total:

The last column in the form is the "Annual Total". Annual total includes service for abnormal days, special services, etc. Annual totals may be calculated by multiplying the total number of weekdays, Saturdays, and Sundays that service was operated by the corresponding data in the respective columns.

4. Unlinked Passenger Trips:

This item is obtained by a two-step process:

a. Divide the sample total number of passengers boarded by the total number of trips in the sample to get the average number of unlinked passengers per trip.

b. Multiply the average unlinked passengers per trip by the total number of bus trips.

5. Passenger Miles:

This is also obtained by a two-step process:

a. Divide the sample total passenger miles by the total number of trips in the sample to get the average passenger miles per trip.

b. Multiply the average passenger miles per trip by the total number of bus trips.

Annual estimates for unlinked passenger trips and passenger miles are obtained by random sampling procedures according to specified confidence and precision levels of 95% and 10% respectively.

The other variable items are self-explanatory, as can be seen on the form. At Valley-Metro this report is submitted at the end of the fiscal year.

Suitable sampling method should be adopted to meet the prescribed precision and confidence level. UMTA Circular 2710.1 describes the sampling procedures for obtaining fixed route bus operating data. The circular details a recommended sampling technique for collecting demand responsive service operating data.

A transit system may adopt a different technique than that recommended by UMTA, so long as it meets the prescribed precision and accuracy levels. If a different technique is used, the methodology for collecting data should be described in the first year's report.

4.2 TIME POINT TRIP FILE

As the name suggests, this file contains separate records for each trip sampled. This too is obtained from the raw data obtained from first stage. A trip is defined as follows "a one-way trip in revenue service beginning at one terminal point (or turnaround point) of a route and ending at another terminal point (or turnaround location) along the same route" (2, p.64)

The Cove Road/Prospect Hills Route thus has two terminal points, the former name as one terminal point and the latter the other terminal point. The Time Point Trip file contains the data and derived information relative to each trip sampled.

Processing, to create the file, resumes at the start of the trip. Each data record is then examined sequentially until the first intermediate time point is identified. During this examination, the stop statistics for the trip between the time points are computed.

At Valley-Metro, the time points are taken to be stops where the scheduled time is defined. Therefore, a time point is defined as "a designated point on a route where specific vehicle arrival and/or departure times are scheduled" (2, p. 64). Some transit systems refer to the time point as a node.

Between time points statistics are calculated for each route. Time Point Trip file could be created for each calendar date; for each route direction or for each trip. In this file, the schedule deviation is also seen. From this schedule deviation, one can know the time lag between the scheduled time and the arrival time.

Schedule deviation at time points is an important factor for deciding the service needs. Information regarding the maximum passengers between time points and the stop location for maximum load is also contained in this file. This information is useful for the transit management in determining at what point on the trip or route the bus had maximum load. With this factor the headway between buses on the same trip or route could be determined. Then the management could decide during what time of the day the headway should be less than other times.

From the schedule deviation the schedule adherence of the bus on each trip or route can be observed. Accordingly decisions can be made in the office to provide on-schedule service. If the deviation is too large it is necessary to know the reason for the late arrival at the particular stop. The management can thus know whether it was a bus failure, operating delay or delay due to large number of passengers getting on and off. A bus failure means that the buses are not maintained properly. Operation delay might include delay in pull out in the morning, traffic congestion on the route or delay due to traffic signals.

At Valley-Metro, scheduled time is known at eight stops on a round trip of the Cove Road/Prospects Hills route. The file created is for the whole day on a particular route at Valley-Metro.

4.3 SOFTWARE DESIGNED FOR SECTION 15 REPORT

The following section describes in detail the computer program that was developed to obtain the Section 15 Report.

As mentioned before, this report is developed from the raw data made available from the output of the first stage computer program. To explain better how the program works, see Figure 10, which is a general flowchart of the program.

The program starts with the MAIN program which reads the 1irst line of the raw data file, henceforth named 'Bus File'. This first line contains the date, month, year, day of week (as a number i.e., 1 = Monday, ..., 7 = Sunday) and most important route number. This route number is used to identify the particular route on which the bus was plying on that

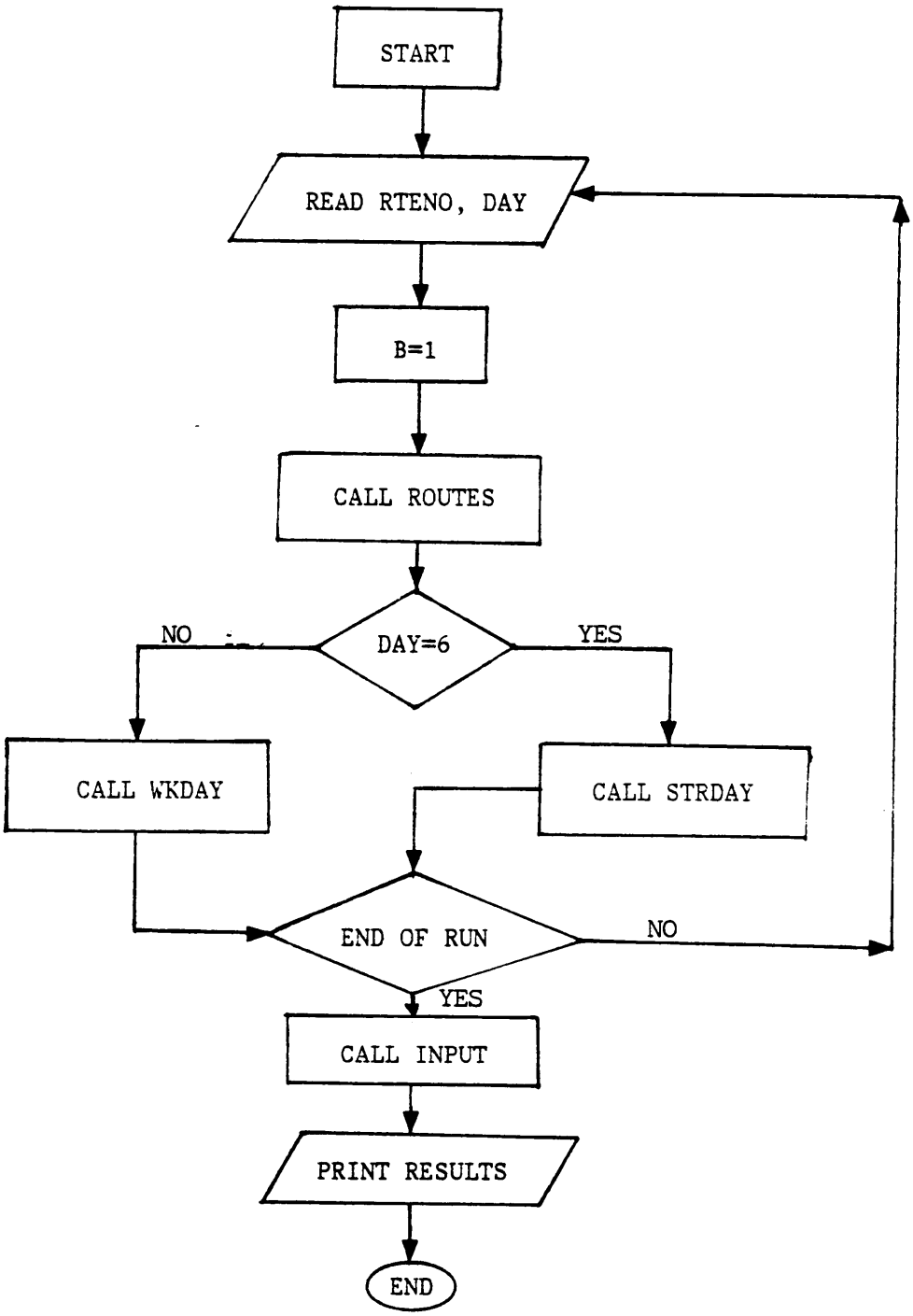

Figure 10. Section 15 Program Flowchart

day. Division by day of week is important since the Section 15 report contains records according to weekday, Saturday and Sunday. At Valley-Metro, buses do not operate on Sundays.

After identifying the route number, the program enters into the subroutine ROUTES. According to the route number the name of the route is displayed on the screen. Then the user is asked to input the number of stops in the route identified. All of the processing begins after the identification of the route number. At the same time the weekday number is also identified. If the weekday number is 6 then it means that the day was a Saturday, as a result the program enters the subroutine STRDAY. If the day was not a Saturday, it enters the subroutine WKDAY.

In the WKDAY subroutine, all the required variable items of the Section 15 report are calculated. An average weekday is divided into four time periods.

1. AM Peak - starting at 6 am and ending at 9 am
2. Midday - starting at 9 am and ending at 3 pm
3. PM Peak - starting at 3 pm and ending at 7 pm
4. Other - any owl or special services after 7 pm and before 6 am.

For each of the above time periods the different items listed in the second column of Section 15 report are calculated. The passengers boarded is directly obtained from the bus file. All the ons in the bus file, between the time points, are added to give the total passengers who boarded. The passengers on board is the passenger load during the par-

ticular time period or day. Again, here too, it is calculated for weekdays, Saturdays or Sundays, if any.

Bus trip distance is the mileage the bus has covered for the day. In other words, it is the vehicle miles that the vehicle has traversed in a day.

Passenger Miles is the next item to be processed in the report. This is an important item in the Section 15 report. The UMTA places special emphasis on this item. Passenger Mile is the distance each passenger travels in a vehicle. Actually it is difficult to calculate the passenger miles since one cannot know which passenger travelled till what stop and also where the passenger boarded is not known. This is very difficult for manually counted data. UMTA suggests to average the number of passengers and multiply it by the distance travelled by the bus in a day. But this gives the approximate passenger miles on a route. In an APC system, it is possible to calculate the passenger miles as close to the actual passenger miles. This is just one of the advantages of an APC system.

The next two items, capacity miles and seat miles are directly obtained by just the knowing the capacity of bus (including standing) and the number of seats in a bus. For practical purposes the capacity, of a bus at Valley-Metro is taken as 60 and the seating capacity as 40. These numbers could be changed if the values are different from the above. This can be done at the beginning of the program.

At the end of each round trip, the program returns to the MAIN program to check if the route is different from the previous loop. If it is different from the previous one, then it re-enters the subroutine ROUTES,

again displying the name of the new route. At the same time the user is prompted to input the number of stops in this particular route. After this is done, it checks the day of the week, if Saturday it enters the STRDAY subroutine. If the route and day are not different from the previous loop it enters the subroutine WKDAY, processing the data till the end of this loop. Thus it goes on till the end of the data is the bus file.

The next important item in this report is the 'Trips in Sample.' At the end of all the data, the total trips is also calculated. This gives us the trips in sample, since it is difficult to collect data for all trips on all the buses. Suitable sampling procedures should be adhered to, to obtain the correct amount of trips to be sampled, which represents for the system as a whole.

If the weekday was a Saturday, it enters the subroutine STRDAY, wherein the different variables for the report for a Saturday are calculated. But the calculations are done for the whole of Saturday, instead of for different time periods as in the average weekday. The reason for this might be the low ridership on a Saturday.

After reaching the last data on the input file, the subroutine INPUT is entered into. In this the user is asked to input the total number of trips for the fiscal year during the AM peak, Midday, PM peak, Other and Saturday. The user should know beforehand the total number of trips on an average weekday - for AM Peak, Midday, PM Peak, Other; for Saturdays and Sundays, if any. Section 15 reports requires the data for the above items for an average weekday, Saturday and Sunday. Therefore, the items are divided by the number of days the data was collected to represent an

average weekday, Saturday or Sunday, as the case may be. Next, the user is asked to input the date of the fiscal year end.

The sample averages of the unlinked passengers per trip and passenger miles per trip are calculated. The annual totals for the unlinked passenger trips and passenger miles are also calculated after all the data in the bus file is screened through.

The computer output is similar in format to the form requested by the UMTA (fig. 11). As mentioned earlier this report is submitted to UMTA, annually, by Valley-Metro.

4.4 SOFTWARE DESIGNED FOR TIME POINT-TRIP FILE

Just as the Section 15 Report is obtained from the bus file, this file/report is also obtained from the bus file. This report contains the statistics of a trip, for a route or for a whole day. It contains records between the time points. As mentioned earlier in the chapter the stops where scheduled time is known is considered as a time point.

In the bus file, we can find the stop numbers, the arrival time, the scheduled time, number of ons at the stop, the number of offs at the stop and the load of each stop. With all these data already known, it becomes easier to create the Time Point-Trip File. Figure 12 is a general flowchart of the program.

At the outset, the route is first identified. The screen displays the name of the route according to the route number identified at the top of the bus file. The user is next prompted to input the number of stops in this route. Thus processing, as explained later, is carried out till

Figure 11. Section 15 Report

Form 406A
ANNUAL REPORT TO UMTA

Transit Id 3007
Fiscal Year End 12/ 3/85

Non-Rail Modes

Mode Level MB R

Line no.	ITEM	WEEKDAYS AM PEAK	WEEKDAYS MIDDAY	WEEKDAYS PM PEAK	WEEKDAYS OTHER	SATURDAY	SUNDAY	ANNUAL TOTAL
	Accumulation From Daily Record Sheet							
01	Passengers Boarded	50	181	63	0	0	0	0
02	Passengers on Board	508	7013	4969	0	0	0	0
03	Bus Trip Distance	27	74	32	0	0	0	0
04	Passenger Miles	371	4067	3374	0	0	0	0
05	Capacity Miles	1620	4440	1920	0	0	0	0
06	Seat Miles	1080	2960	1280	0	0	0	0
07	Trips in Sample	2	4	2	0	0	0	0
08	Total Number of Bus Trips	0	0	0	0	0	0	0
	Sample Averages							
09	Unlinked Pass. per Trip (1/7)	25	45	31	0	0	0	
10	Pass. Miles per Trip (4/7)	185	1016	1687	0	0	0	
	Annual Totals							
11	Unlinked Passenger Trips (8x9)	0	0	0	0	0	0	
12	Passenger Miles (8x10)	0	0	0	0	0	0	

the route changes. If the route is changed, again the screen displays the name of the route. The user then has to key in the number of stops in this particular route. Processing, thus, continues till the end of the data.

The first time point is initially identified. After this identification the between time point statistics in calculated. By between time point statistics is meant the ons and offs; the arrival and schedule times are also noted.

The next step is to identify the next time point. The above process is continued till the next time point is identified. Cumulative statistics is continuously calculated till the next time point.

At the same time at every time point, it is checked whether the bus arrived at the stop or not. If it did arrive then, the schedule deviation of the time is calculated, i.e., the deviation of the arrival time from the scheduled time. As a convention if the arrival time is more than the scheduled time, the deviation is negative. A negative deviation means that the bus arrived late at that stop, than scheduled.

Not only is the schedule deviation an important factor of this file, but the maximum load at any stop between time points is also an important factor. Maximum load is obtained only at stops where it has arrived, i.e., load is obtained only when the passengers get on or off.

As can be seen this file contains separate records for each route direction. But if necessary for each calendar date, this file is created this way. Again the program, if the day is changed than the previous, the user is asked whether processing is to be continued or not. If yes, it will add up the statistics to previous days statistics and continue

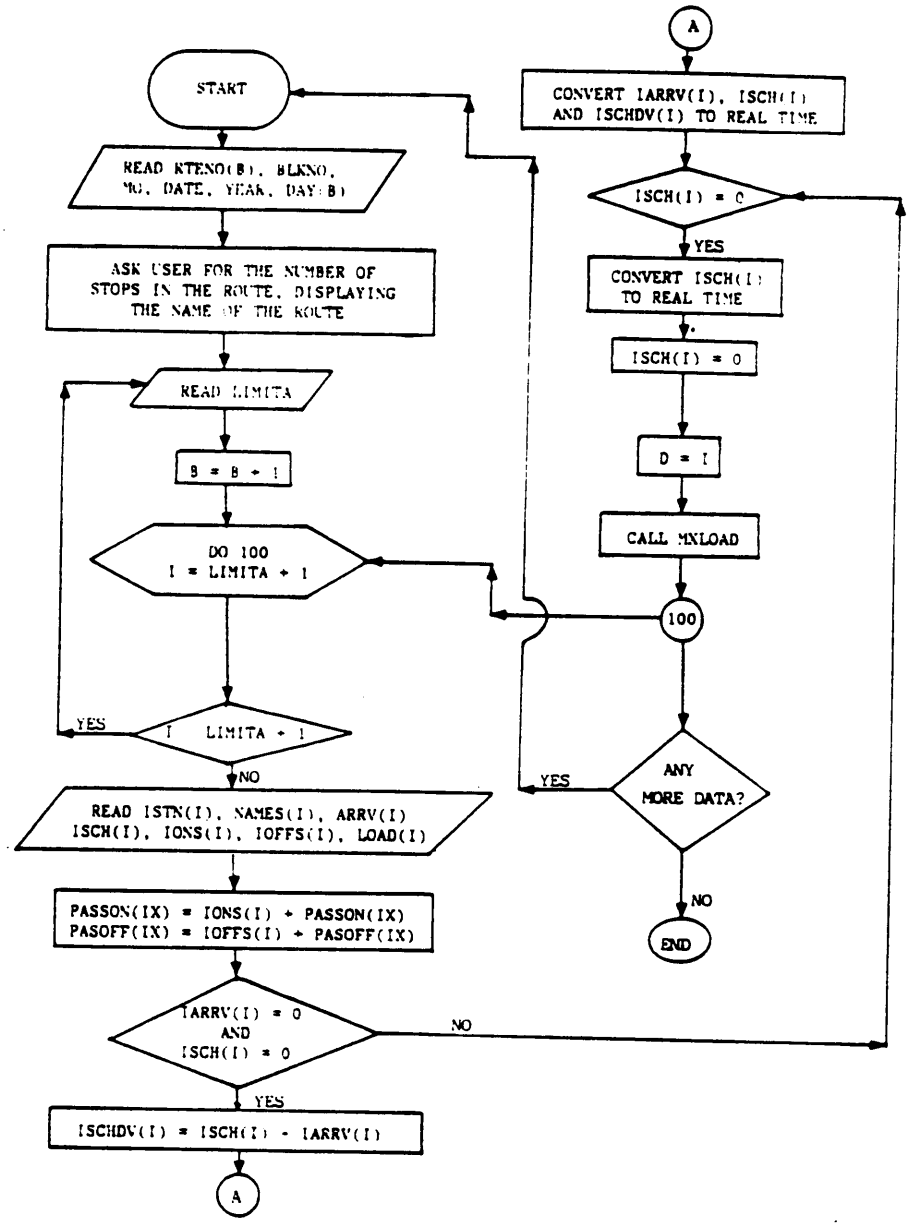

Figure 12. Time Point Trip File Program Flow Chart

processing till it hits a different date or route. This program is thus flexible for any route at Valley-Metro.

Finally, it constructs a bus profile file and stores the data into this file called the Time Point-Trip file. Figure 13 is a part of the Time Point Trip File.

This file helps the transit managers at Valley-Metro to know how late the bus is running on a particular route. Schedule deviation if positive means that the bus is ahead of time, whereas a negative deviation means it is running late. If might be late due several reasons. Delay on part of the operator early on the morning, during pull out, might be one of the reasons. Also the maximum load between two time points is a pointer for service needs. If it occurs everytime on the same route during the same hour, then the management should think of providing more buses if the load is affecting the arrival times. But at Valley-Metro, from information gathered at their office, the ridership is not a factor that may affect the service of a bus. The service needs could be decoded accordingly by studying the time point-trip file.

These two reports are very important for the Valley-Metro. Section 15 report is one that is useful to them since it has to be submitted annually to UMTA to get more grants for the transit company.

Figure 13. Time Point Trip File

ROANOKE VALLEY-METRO APC SYSTEM

ROUTE 1 BLOCK 17 DATE 12/19/85

STOP #		ARRIVE	SCHED	SCHDEV	PASS ONS	BET PASS OFFS	TIME PTS	LOAD
1	BRENDLES (Time Point)	10:59:15	10:55: 0	- 0: 4:15	25	24		12
34	ROANOKE MEMORIAL HOSPITAL	11: 9:45	11:10: 0	0: 0:15	3	0		15
49	JEFFERSON & CHURCH	11:22: 0	11:20: 0	- 0: 2: 0	2	4		13
67	ORANGE & 10 th		11:28: 0		7	0		20
106	FERNCLIFF APARTMENTS N	11:44:45	11:40: 0	- 0: 4:45	8	3		18
144	ORANGE & 10 th		11:57: 0		7	4		21
166	JEFFERSON & CHURCH	12: 8:30	12:10: 0	0: 1:30	20	8		30
181	ROANOKE MEMORIAL HOSPITAL		12:17: 0		3	0		33

STOP NO = 34 MAX LOAD = 15

STOP NO = 49 MAX LOAD = 13

STOP NO = 58 MAX LOAD = 20

STOP NO = 70 MAX LOAD = 19

STOP NO = 112 MAX LOAD = 23

STOP NO = 166 MAX LOAD = 30

STOP NO = 170 MAX LOAD = 33

STOP NO = 183 MAX LOAD = 34

5.0 COMPUTER GRAPHICS AS DECISION AIDS

Computer generated graphics are becoming increasingly popular and easily available to decision makers. It has been claimed by all users, that the use of graphics improved decision speed and quality over traditional methods of data display. Correct interpretation of graphical displays require training, which most users lack.

Despite the vendors pushing the computer users to purchase graphics products, relatively little is known about the actual utility of graphics as decision aids. According to proponents, graphics meaningfully improve managerial productivity (14, p.463). In a recent discussion of the topics, Ives concluded that "the failure to demonstrate a clear advantage for graphics suggests that the extravagant claims favoring graphic presentation formats may be considerably overstated" (14, p.463).

The most common questions about graphics being used to communicate information to decision makers are: how do graphical decision aids affect organizational decision making? What happens when information that traditionally has been communicated in tabular form is presented in graphical form? Thus, graphics is a tool for Decision Support System (DSS) for the decision makers.

According to Keen and Scott-Morton, a Decision Support System "focuses on managers" decision-making activities and needs while extending their capabilities." Furthermore, DSS implies the use of computers to

- Assist managers in their decision processes.

- Support, rather than replace, managerial judgment.
- Improve the effectiveness of decision making rather than its efficiency (14).

In short, the purpose of a DSS is to provide support for the judgment of one or more users in their principal function of making decisions.

Graphics may be used in organizations to serve a variety of purposes:

1. To facilitate the design of equipment and facilities, usually in the manufacturing area of the form (design graphics)
2. To organize or schedule activities for planning and control purposes (scheduling graphics)
3. To analyze data as part of a statistical or financial analysis (analytical graphics)
4. To support the decision-making activities of high-level management (decision graphics).

The second purpose is the important use for the management/decision makers at the Valley-Metro. To schedule the activities for planning their bus routes, headways, etc., graphics is an important tool for supporting their decisions. Keen and Scott-Morton have mentioned that an effective decision support tool is one that results in greater decision effectiveness or efficiency (15, p.467). For graphical presentation, various softwares are available. One such software is the Lotus Symphony, developed by the Lotus Development Corporation, Cambridge, MA.

5.1 LOTUS SYMPHONY

What is Lotus Symphony? Symphony is a rich and flexible program that combines five capabilities in one package:

1. Word Processing: A powerful and versatile tool that helps professionals write letters, reports, memos, and other business correspondence.
2. Spreadsheet: A tool that performs the sophisticated numeric and financial analyses needed for planning and decision-making.
3. Business Graphics: A tool that converts numeric data into graphs and charts for analyzing financial information and visually communicating the results to others.
4. Database Management: A tool for storing, organizing, and managing information electronically. This information can be selectively retrieved and used as the basis for analysis, discussions, planning, and decision-making, and for generating reports, form letters, and mailing lists.
5. Communications: A tool for exchanging information with people using other computers and for assessing electronic bulletin boards and public databases (16).

Symphony provides five work environments in which you use its capabilities. Each environment has unique features and commands. The work environments -- Sheet (spreadsheet), DOC (word processing), Graph (business graphics), Form (database management), and Comm (communications) are

different from each other. With a single keystroke, you can change from one environment to another. You can use Symphony's capabilities alone or together to accomplish your objective, and you can maximize Symphony's power further by using its special features.

For the plots and graphical displays at Valley-Metro, only the spreadsheet and graph environments were used. The sheet environment is used to store the output of the simple fortran program. Then by suitable manipulations, the graphs are developed that will be displayed on the microcomputer screen.

The Lotus-Symphony software is contained in six disks:

1. Symphony Program Disk
2. Help and Tutorial Disk
3. Install and Utility Disk
4. Master Library Disk
5. Tutorial Lessons Disk
6. Printgraph Program Disk

The utility of each disk can be found in the Symphony manual obtained along with the disks. To develop the plots/graphs for Valley-Metro, the main disks used are the Symphony Program and Print Graph Disks.

One of the most important feature of the Lotus-Symphony is the macro facility. You can create macros that replay keystrokes you frequently use. For example, as you write a report, you might type a client's name over and over again. To avoid doing so, you can create a macro that types the name for you. When you create the macro, you associate it with a

particular key. After that, you can press the key at any time to insert the client's name. In creating the plots for Valley-Metro, this facility is used, i.e., by pressing 3 to 4 keys, a whole series of commands are executed, the final output being the graphic display of each plot. This facility saves a lot of time and energy thus speeding up work for the management. The macros developed for each plot is explained in detail in the subsequent sections.

5.2 UTILITY OF LOTUS SYMPHONY AT VALLEY-METRO

As mentioned earlier, the extensively used environments of Symphony are the Sheet and Graph environments. Basically it was decided to create graphs of three types:

1. Route Demand Plot and Route Demand Bar Plot
2. Passenger Load Plot
3. Load Profile Plot and Load Profile Bar Plot

A brief description of each plot is mentioned below:

1. Route Demand Plot:

This graph illustrates the plot between the total passenger demand and the time of day. It shows the number of passengers that boarded during every hour of the day. This report illustrates the demand for service on a given route as evidenced by the number of people who boarded the buses during each hour of the day.

2. Passenger Load Plot:

This plot shows the load at every moment of the day. With the APC system, it is possible to determine the passenger load at every moment of the day. Both the magnitude and duration of the loads can be observed in this graph/plot.

3. Load Profile Plot:

This plot gives the load at the different time points along the route. It might be obtained for a particular time of the day or for the whole day.

The above plots will help the management at Valley-Metro to improve the efficiency of the service, after decisions are made by the management.

6.0 ANALYSIS OF THE PLOTS

As mentioned earlier, the APC is an important tool for transit properties. Its main purpose is to provide the transit management with current, accurate and meaningful statistics regarding the transit operating performance and the level of service offered. There is a continuous requirement by the management for this information as the basis for route planning, route design, service scheduling, operations forecasting and financial budgeting.

Though all of the above reports could be generated from the APC with adequate information, only the three previously mentioned plots were required to be generated as part of this thesis and stage of work. These three plots fulfill a part of the management requirements to support their decisions. From these reports it is possible for Valley-Metro officials to have a knowledge of the passenger trend on the different routes and also help them in their scheduling activities. Depending on these plots, however, major decisions cannot be undertaken, since they provide little information regarding the actual trend of the boardings on the buses on different routes.

6.1 ROUTE DEMAND ANALYSIS

As mentioned in the previous chapter, the route demand plot displays the total passenger demand during the time of day (fig. 14 & 15) By time

of day is meant all the 24 hours. Buses at Valley-Metro start plying no later than 5:40 am in the morning and are back into the barns by 7:00 pm.

The Route Demand Plot illustrates the demand for service on a given route as evidenced by the number of people who boarded the buses during each hour of the day. From this plot, one can know the exact number of passengers boarded during any hour of the day. The plot could be obtained for a certain amount of days say for a month or for a particular season. During winter season, it has been observed that the ridership is less on most of the routes. At a glance, one can observe the trend of ridership on the route under consideration.

Also, the ridership for time periods could be checked. Say for AM peak and PM peak the ridership is high, then the headways between the buses could be reduced. This would mean plying more number of buses on the route. If the passenger occupancy is high, more buses are required to meet the demand.

The graph could be obtained for a weekday or a Saturday also. On the same graph, one can observe the change in ridership on a Saturday or on a weekday. Thus, the scheduling department at the transit company could alter their schedules according to the demand of ridership. They can also, infer by judgment the number of buses required during peak and off-peak hours. This would save a lot of unnecessary running of the buses, at the same time reducing maintenance costs, and minimizing fuel consumption.

The importance of this plot is that, decisions could be made in rationalizing any potential route changes. For example, if on a particular route the ridership is less, then the bus could be re-routed in another

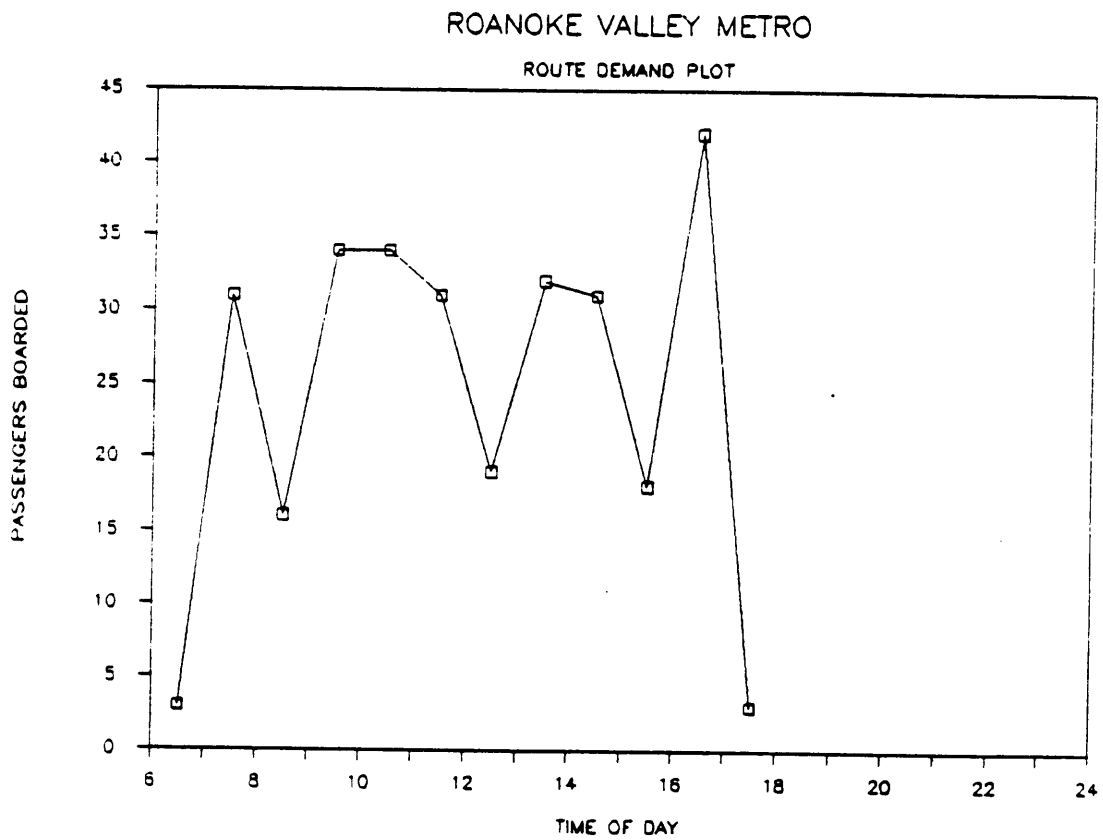

Figure 14. Route Demand Plot

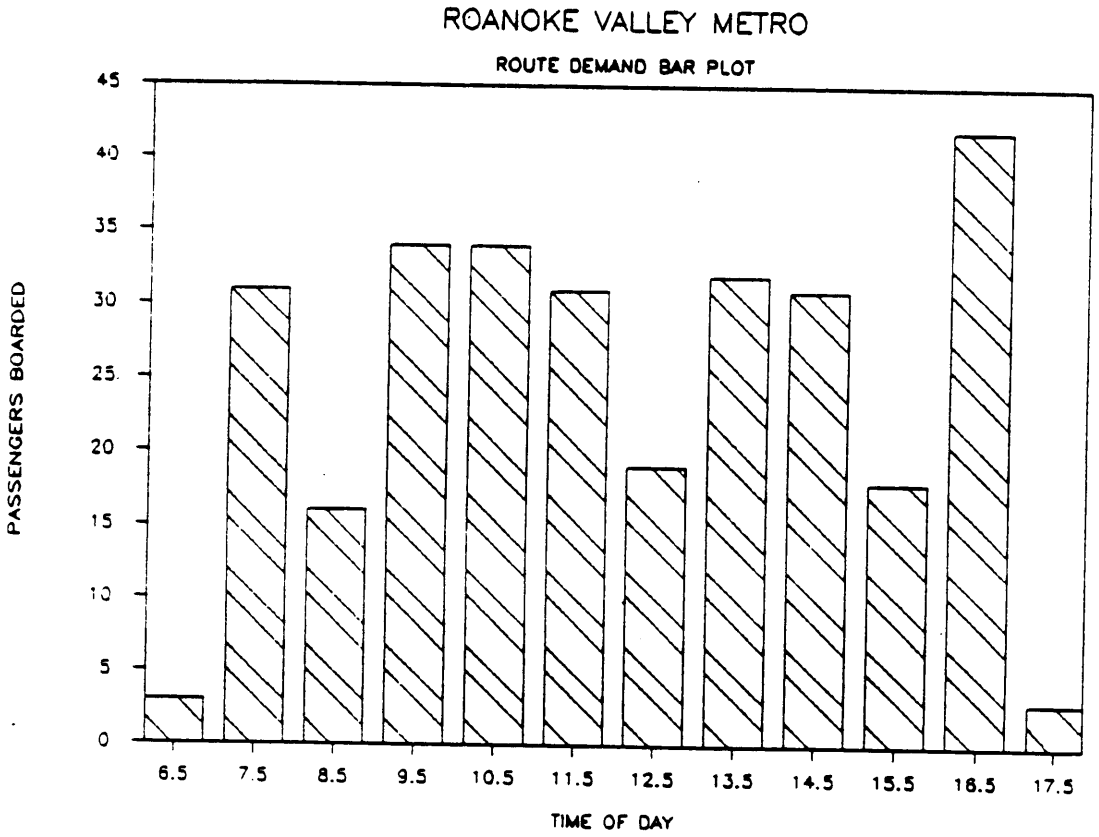

Figure 15. Route Demand Bar Plot

direction. The re-routing would definitely help the riders in the surrounding area, though it might be for a few hours, during a certain time period. With this decision, the transit company could not only earn more revenue, but also be popular among the riders and community at large.

6.2 PASSENGER LOAD ANALYSIS

Ridership varies not only from day to day, but also between time-periods it varies considerably. The load pattern during any time of the day on a particular day of a route could be observed in the Passenger Load Plot (fig. 16). From the APC data, after suitable manipulation of the passengers boarded, the load at every stop or any moment could be obtained. This plot shows the passenger load continually over the vehicle's tour of duty in a day. Clearly one can see that the load during the AM peak and the PM peak is high. During Midday it is low compared to the AM peak.

In this plot, both the magnitude and duration of the loads can be observed. Knowing the magnitude, suitable decisions could be made. If the load was greater than the capacity of the buses, then it could be infered that more number of buses are required in that route, if not at least the headways should be reduced so as to reduce the load on the bus. Along with providing quick transport for the public, it should be seen that they ride comfortably.

This plot is important in the dispatching operation. For example, if it is observed that the load is high on a particular route for a certain

day of the week, then a special bus could be dispatched to satisfy the demand of the riders, i.e., a tripper service could be put in operation.

Ridership depends a lot on the weather and during bad weather people tend not to travel by buses. From the data from APC, it could be seen that the load pattern is different during bad weather, thus requiring less number of buses in operation.

The U.S. Department of Transportation has set some loading standard guidelines like

1. In off-peak periods, all passengers should have a seat.
2. Rides for standees should not exceed 15 to 20 minutes.
3. In off-peaks periods, trips can be eliminated if the average bus loads are 10 passengers or less. (15)

The main objective being that the transit system should stay within load standards. Thus the Passenger Load Plot helps the transit company to follow the DOT (aid guidelines, without any violations). The best suggested guideline to adhere to the above strictures is to establish peak and off-peak loading standards and adjust service to meet the standards. It should be seen that a certain trip has a minimum average passenger ridership. Similarly, for peak trips also the ridership should not be more than the load standard already set.

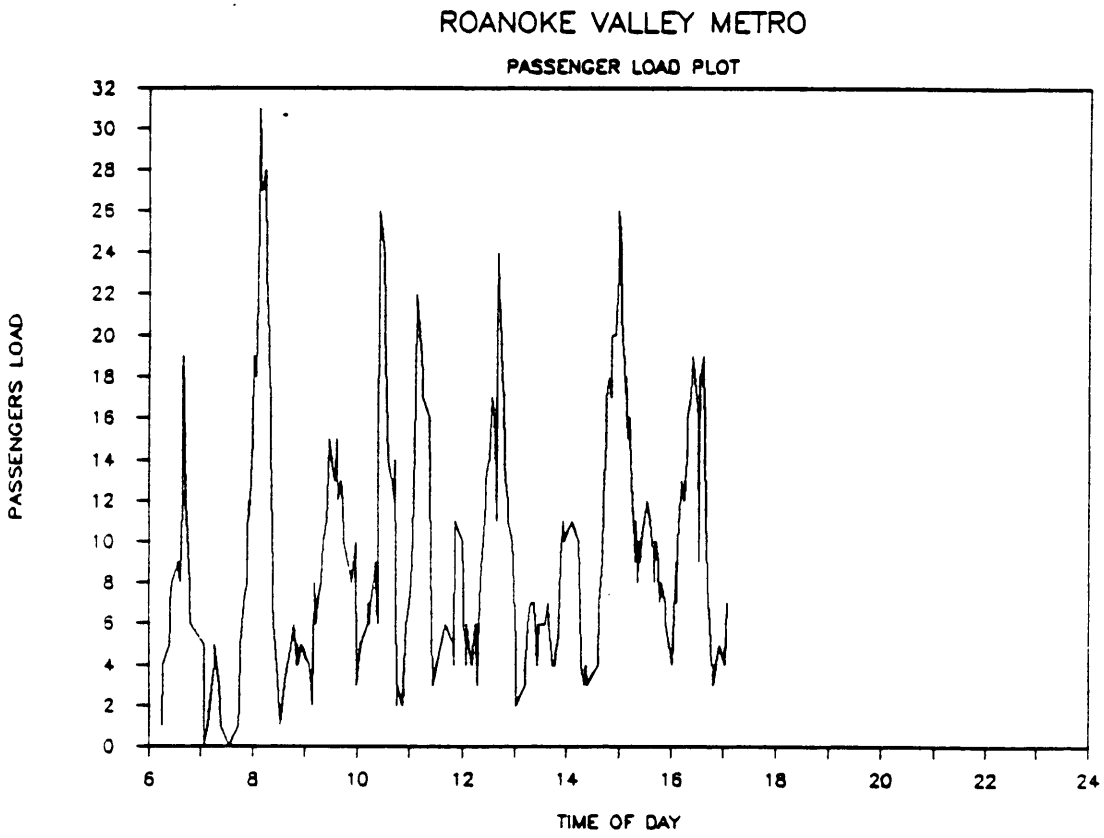

Figure 16. Passenger Load Plot (sample)

6.3 LOAD PROFILE PLOT

This plot shows the passenger load at the time points. The time points are defined as in the Time Point Trip Report. The load at time points gives one an idea of the trend of the passengers alighting. This plot is obtained for a particular route and for a day, either weekday or Saturday (fig. 17 & 18).

The plot could be determined for the different time periods, but since ridership on the Valley-Metro is not very high, it was found sufficient to create a plot for the whole day on a particular route.

From this plot, it could be seen as to which point on the route the bus has maximum load and minimum load. This is useful again for scheduling purposes, so that the transit managers would know at which point the load is maximum. The eight time periods on this route with the stop numbers and the names are as follows:

1. Brendles
2. Roanoke Memorial Hospital
3. Jefferson and Church
4. Orange and 10th street intersection
5. Ferncliff Apartments
6. Orange and 10th street intersection
7. Jefferson and Church
8. Roanoke Memorial Hospital

These time points are on the route for which the APC was tested.

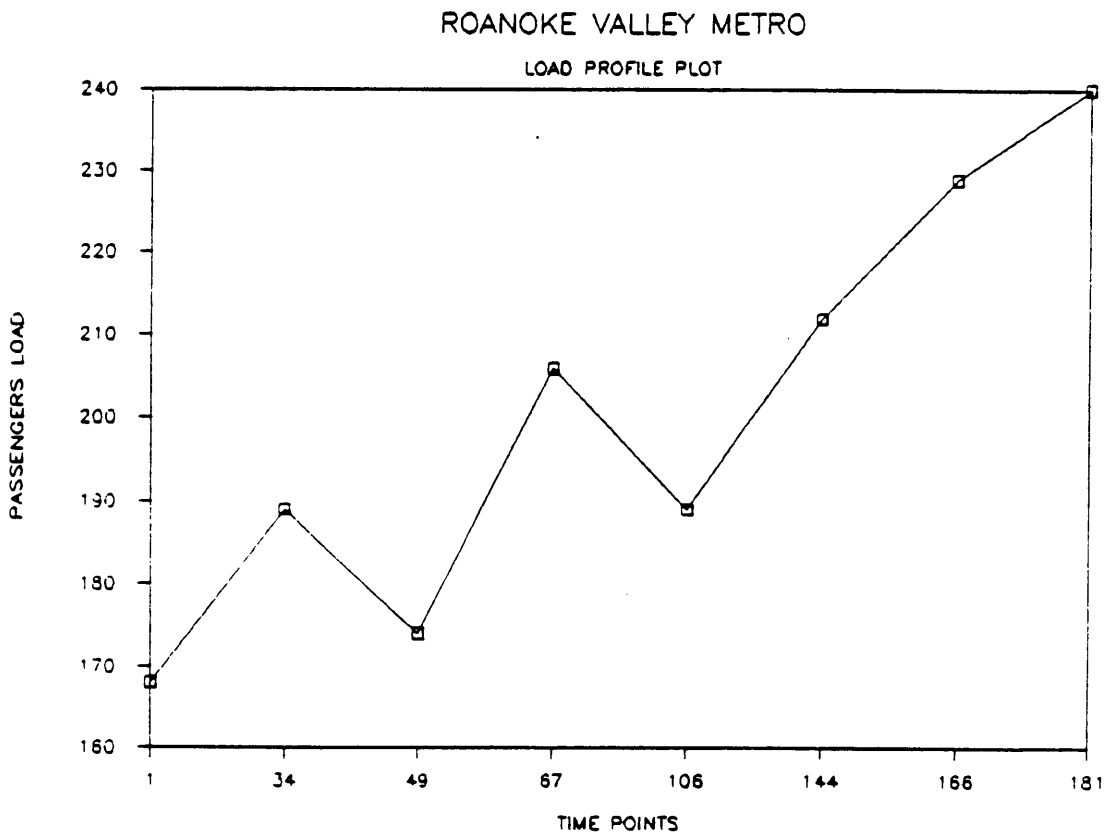

Figure 17. Load Profile Plot

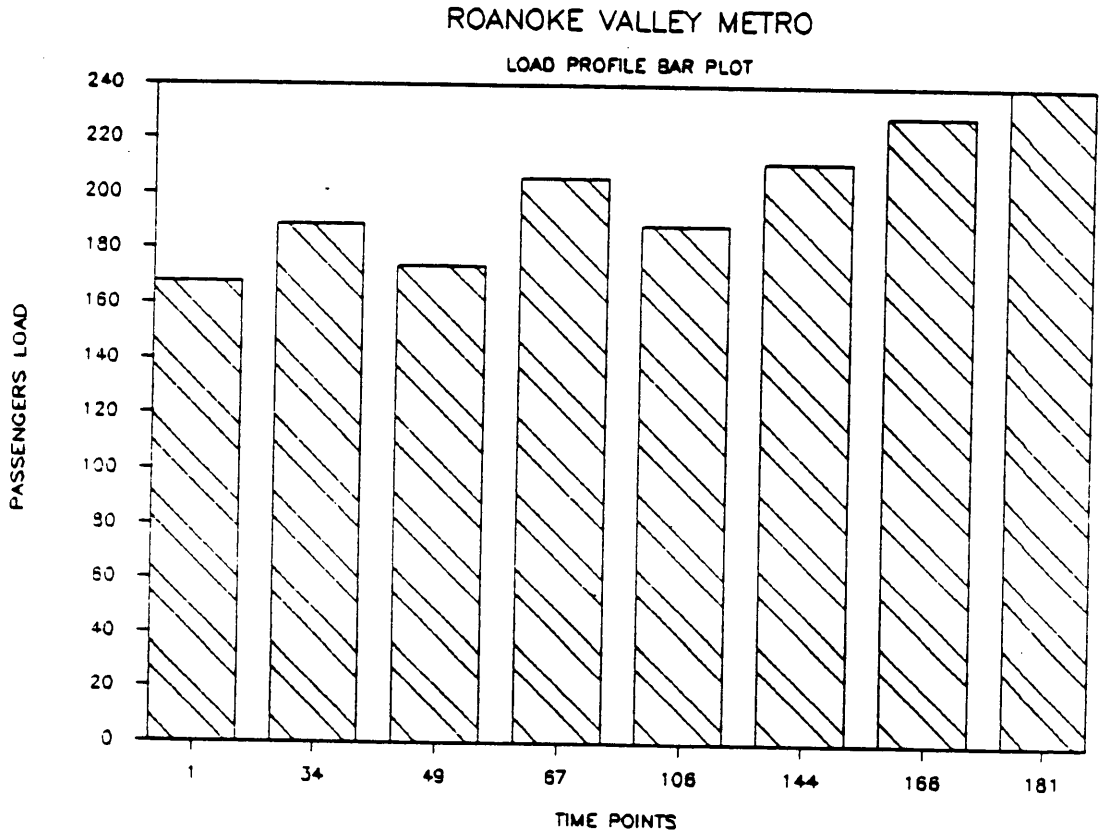

Figure 18. Load Profile Bar Plot

The following sections explain the software programs developed to create the above three plots.

6.4 PROGRAM DESCRIPTION

The plots are obtained in two stages. The first stage being the simple Fortran program developed to obtain the input for the plots that will be created on Lotus-Symphony. The second stage is the program developed in the Lotus-Symphony to create the plots. The Symphony is itself a powerful software program. It has a lot of capabilities as mentioned in earlier chapters.

6.4.1 Description of the Fortran Program

The variables needed for the three plots are as follows:

1. The arrival time at each stop
2. The load at that stop
3. The passenger ons at each stop
4. The time points on the route considered
5. The load at each check point

All the programs developed are compatible to all the routes except for the last part. This part consists of the time point numbers for a particular route. In this case it is the Cove Road-Prospect Hills route.

Just by changing the stop number of time points in this part, it is possible to run this section also.

The basic function of this program is to create the suitable database for the Symphony programs by easy and simple manipulation of the bus file (obtained from Stage 1 of the project).

As in the earlier program, here too the route number is first identified. Once the route number is identified, the subroutine ROUTES asks the user the number of stops on this route, at the same time displaying the name of the route as at Valley-Metro. The user should know beforehand the number of stops on a route.

The first part manipulates the input file to calculate the passenger ons in an hour starting from 6 am upto 12 midnight. This subroutine is important to create the Route Demand Plot. Since this plot evidences the number of people who boarded the bus during each hour of the day, this subroutine calculates the passengers boarded during every hour of the bus run.

The next subroutine calculates the variables required for the Passenger Load Plot. This second part, finds out the load at each stop whenever the bus has arrived. Also, the arrival time, which is in seconds, is converted to hours.

The last part, accumulates the load at each time point on the route. This section could be used only for the Cove Road-Prospect Hills route since in the program, the time points have been defined by specific variables. But, by just changing the time point stop numbers, it is possible to use this section for the other routes. Also the number of time points

on each route is to be considered. On this route there are eight time points, but on the other routes it may be less than or greater than eight.

All the output are finally written into a file. The titles at the top of the output file have to be enclosed within converted commas, so as to make the symphony read them as labels instead of not reading them at all. Names and words have to be enclosed in inverted commas to retrieve them on the symphony spreadsheet.

6.4.2 Description of the Symphony Macros

As mentioned earlier, Symphony is a rich and flexible software. Basically, Symphony is used in this project, to create graphs. Though it has other excellent capabilities, which are not used at this stage, later on the Valley-Metro transit management could use this for their daily use such as scripting, database management, etc.

One such capability is the macro facility available in Symphony. Symphony can perform tasks automatically when you write a set of instructions, called a macro, as its guide. To explain briefly, first, you compose the macro as one or more cell entries. Once you have written and named the macro, you can invoke it. Symphony reads the script and executes the instructions all at once.

Macros are useful in many situations, especially, if one is designing graphs and wish to crate different graphs. A macro reduces this process to a single or a few keystrokes. Automating a keystroke sequence is only part of what a macro can do. A macro can also be a complete program, with processing features such as loops, branch points, and subroutines.

A macro is a series of cell entries arranged in a single column. An empty or numeric cell signals the end of a macro. Symphony stores a macro in the worksheet along with the other data. Once Symphony starts executing a macro, it continues down the column of cells until it finds an instruction that stops it or sends it somewhere else to continue. Whenever the worksheet is saved even the macros on that worksheet are also saved.

One can create macros by typing label entries in a column of cells. Macros should be created in an empty area of the worksheet so that they do not get in the way of the data. The Symphony Reference Manual gives all the details of creating, naming, and invoking a macro. Even the command language with which macros could be developed can be found in the reference manual.

6.4.3 Macros Designed at Valley-Metro

The Symphony software is a program in itself, i.e., it can do various functions by just hitting the particular keys. Thus, the macros should be so designed so as to manipulate the various functions of symphony to obtain the desired results.

To start symphony, the manuals has to be referred to. After Symphony is started user should ask for the file SUNIL•WR1. All the file names in the diskette are displayed on the screen. By just moving the pointer, the file is displayed on the screen, on hitting the return key. To invoke the macro right away, the key, USER (F7), is depressed, and the name of the macro is keyed in. The name for the macro created for the graphs is

PLOT. The name is displayed at the right hand bottom corner as the user keys in. After typing the macro name, hit return. After a series of commands the graphs are displayed.

This macro created is interactive. In the MAIN program, the user is asked if he wants to create a few plots. If yes, then the user is asked whether he has all the data on this file. If no, then he is asked to import all the data and then he has to quit. To import the required data file it is easy, and can be done by using the MENU key.

If the user says he has the data, then he is asked which plot he needs, i.e., route deman plot; passenger load plot or load profile plot. This is executed by the subroutine PLOTS.

In the subroutine FIG1, it starts with the menu. From this menu, the graph command is selected. In this the first settings is selected. In first settings, the type of graphs available is to be chosen. There are different types of graphs available like line graphs, bar graphs, stacked bar graphs, pie charts and XY graphs. For the demand plot, the XY graph is suitable. After XY type is selected, it returns to the previous menu. From this, the range for the X-axis and Y-axis is to be determined. Here the user is required to point the range for the x axis, i.e., the data to be plotted on the X-axis. It is the time of day. But before this during the execution of the main program, in the first column of the spreadsheet, the numbers 6.5 to 23.5 are created. These numbers indicate the hours in decimals, i.e., 6.5 is 6:30 am, and so forth. After the X-range is determined, the A-range is indicated. The A-range is the data along the Y-axis.

Next, the menu switches to the second settings. The second settings contains the facility to naming the plots, including the type of data along the X and Y axis. The Y-scale and X-scale, naming the plot and other functions such as grid, color, etc., are available.

After switching over to second setting's menu, the X-scale is determined. The X-scale could be either automatic or manual. For this plot manual-linear is selected which then asks for the lower limit and upper limit for the X-axis. Since the time of day is plotted on the X-axis, and buses begin to ply after 6 am, the lower limit should preferrably be 6 and the upper limit 24.

Next the Y-scale is selected. For the Y-scale, the type selected is automatic since, Symphony will automatically determine the lower and upper bounds depending on the minimum and maximum values of the data in the A-range. From the same menu, the width for the points on the Y-scale is selected. Symphony by default offers a width of 9, but a width of 3 gives out a clear plot.

Quitting from the above menu, Symphony asks whether we require a color plot, selecting yes, the next step is the titles for the plot. The first title is the name of the bus system, Roanoke Valley-Metro (it can be anything else). The second title is the title of the plot, for example, Route Demand Plot. The title along the X-axis for the above plot should be the time of day and that along the Y-axis, Total Passengers Boarded.

The next macro execution is the preview of the graph. The plot is displayed on the screen. The whole procedure starting from invoking the macro to the display of the graph takes about one minute.

Previewing the plot, hitting return, the user is asked whether he wants to save the plot. If yes, the plot is given a suitable name and is saved, so that the plot is printed in the Print Graph environment.

Exiting from this subroutine, it returns to the Subroutine PLOTS, and asks the user if he wants the Passenger Load Plot. If the user selects yes, then it enters the subroutine FIG2. This subroutine is capable of plotting the Passenger Load Plot. The logic behind this subroutine is the same as in FIG1 subroutine. Most of the commands and manipulations are similar to the first one except for a few minor changes.

From this subroutine it moves back to the PLOTS Subroutine, asking the user whether he wants to view the third plot, Load Profile Plot, thus entering the Subroutine FIG3. This subroutine is also quite similar to the other two, except for some changes.

The important thing to be noted here is that the X-range and A-range for the three plots are different, so one has to be careful while selecting the ranges. Macros could be created for any function, thus making things easier and faster.

7.0 ACTUAL INSTALLATION AND OPERATION

7.1 PLACEMENT OF THE EQUIPMENT

The four modules were installed in the bus at the back of the front seat away from the sight of any passengers. At present there is no back door module since it had some problems in registering the offs from the back door sensors. The two pairs of sensors at the front door are fixed at an average height of 48 inches from the inner footsteps of the bus. The sensors on the same side are skewed at a height of six inches as well as a width of six inches. This is done so that the sensors do not miss a person too thin or too fat.

The sensors at the back door are placed in a similar fashion as the front sensors. The back door sensors get activated whenever the green light above the door is lighted. The front door sensors are activated when the front is opened.

7.2 ACCURACY OF THE APC

To determine the accuracy of the APC, manual checking was conducted simultaneously with the APC switched on. This checking was conducted for about 15 loops. Each loop is about 20.72 miles long and to complete one loop it takes about one and a half hour. Two persons actually sat on the bus to conduct the survey. One person checked the actual ons and offs and then noted them along side the arrival time at that stop. The other

checked the logs immediately on the Data General, so that it was known whether the APC registered the correct ons and offs. The diagnostic unit was also used to check the time and distance and the ons and offs.

In the first five loops collected, a few errors were found in the counters, for example, distance was not recorded correctly, the passenger logs were being registered after the bus moved for three minutes or more. Due to these reasons matching of the stops were being mis-matched. This was basically a hardware problem. The old chip in the storage module was then replaced with a new chip with the necessary changes in it. The distance that is registered by the logs was supposed to record at an odometer impulse of approximately 48 feet. But after checking the data from the logs and from the diagnostic unit it was found that the odometer impulse was approximately 11.6 feet. Also the passenger log was being recorded after the bus moved for three minutes or more. On the Cove Road/Prospect Hills route, the stops were very close to each other. Some stops are within two minutes distance. Hence, during the matching of the data, there used to be an error in matching. Instead of getting matched to a particular stop, it used to get matched at the next stop, thus giving the statistics at the wrong bus stop. This error was also due to the software in the hardware chip. After rectifying this, it was seen that for the next ten loops the matching was good. Now, sometimes, there is some mismatching because of the same problem. In the ten loops this occurs at least once in each loop. This occurs only if the passenger log (05) is recorded after two minutes of the bus movement. This log is then matched to the next stop. This error, though, is not quite frequent.

In the ten loops, from which manual count was also done, the percentage error for ons was found to be 6% whereas for the offs it was found to be 5%. In the ten loops, (each loop has 209 stops) error in ons occurred at 24 stops, whereas, error in offs occurred at 20 stops. Therefore, the probability of an error occurring at each bus stop for ons = 1.6% and offs = 1.1%.

The average difference in the distance measured by the APC and the actual, for the ten loops was found to be 0.25 miles, i.e., about 1.2%. The time recorded was found to be correct without any variation.

7.3 RETRIEVAL OF THE DATA FROM THE CSM

At present, the data from the CSM has to be retrieved everyday since the CSM is sets all the data to zero once the power is put on. The reason for this being, power is supplied from the battery of the bus. So only when the bus battery is on, the CSM is working. Later it was suggested to place a battery that will make the CSM store the data at least for a week. This is quite convenient instead of dumping the data everyday. This way the data could be dumped every week.

The Data General One is first connected to the CSM with the microfloppydisk (3.5 inch disk) in the A drive. The Data General One is a portable microcomputer which could be conveniently taken on the bus. The Data General is battery charged and would operate for eight hours continuously. When the microcomputer is switched on it goes into basic mode and then the 'OK' is displayed, which means the system is ready. At the bottom of the screen one can see all the functions that each

	ONS		OFFS	
	MANUAL	APC	MANUAL	APC
DEC 17,'85 (3 loops)	143	131	81	82
DEC 19,'85 (2 loops)	100	91	47	43
DEC 24,'85 (2 loops)	70	67	58	56
JAN 3, '85 (3 loops)	136	133	90	81
TOTAL	449	422	276	262

TOTAL PERCENTAGE ERROR (for the 10 loops):

ONS = 27/449 OFFS = 14/276

= 0.06 = 0.05

= 6% = 5%

PROBABILITY OF AN ERROR OCCURING AT EACH BUS STOP:

FOR ONS = 1.6%

FOR OFFS = 1.1%

Figure 19. Error in Ons and Offs

ACTUAL DISTANCE OF ONE LOOP = 109418 ft. = 20.72 miles

ERROR IN DISTANCE MEASURED (at an odometer impulse of 11.6')

DEC 17,'85 (2 LOOPS) -----	0.37 miles
DEC 19,'85 (1 LOOP) -----	0.23 miles
DEC 24,'85 (4 LOOPS) -----	0.25 miles
JAN 3, '85 (3 LOOPS) -----	0.19 miles
TOTAL	1.04 miles

TOTAL PERCENTAGE ERROR (for the 4 days) = 1.2%

Figure 20. Error in Distance

function key would perform. The various steps to dump the data are as follows:

1. First hit the LOAD key (F3 key)
2. The cursor appears next to the word LOAD. Type in NEWONE and hit return.
3. Ok, appears on screen.
4. Hit RUN (F2 key)
5. A menu is displayed
6. Select T to transfer data from CSM to Disk. (The letter selected should be in capitals) and hit return.
7. A sentence displaying:

   S-Screen only, D-Disk only, B-Both. Enter choice .... is seen on the screen.
8. If you want to see on the screen while it is being dumped onto the disk, select B, else D. Selecting S displays the logs on the screen only.
9. After step 8 is completed, the user is asked to input the real time in hours and minutes. This is important since backtagging of the data starts from this time.
10. After keying the time, the user is asked to input the Bus ID, the month, day, year, day of the week and the initial of the user.

These ten steps describe the retrieval of the data from the CSM on the 3.5 inch micro floppy disk. The next section describes the process of dumping the data from this microdisk to the 5 inch floppy disk, so that

it could be used on the IBM PC to develop the reports from the software program.

1. Insert the microfloppydisk into the portable C drive.
2. Connect the portable C drive to PC
3. Insert the MANDRV disk in the A drive of the PC
4. Insert the disk onto which the data is to be transferred in drive B
5. Boot the system
6. The A> prompt will appear. Type in C:. The C> will now appear on the screen.
7. Type COPY FILENAME FILE EXT. B:
8. The file has now been copied on the disk in drive B.

7.4 STEPS TO EXECUTE THE PROGRAMS.

To develop the outputs for each route, the two external files, i.e., the Distance file and the Schedule file should be on the disk. These external files must be developed for each route to obtain the report for each route, if the data collected is for the particular route. The distance file contains the distance between the stops in feet. For the Cove Road-Prospect Hills route the first stop is considered to be the stop at Brendles in Tanglewood Mall, Roanoke. For convenience sake the last stop is considered to be the signboard record, wherein a log is created when the operator of the bus changes the signboard to indicate the direction of the route. The schedule file contains the scheduled time in seconds for each of the time points. On this route there are eight time points.

The format for each of the external files should be similar to the one shown in Appendix A.

To start with the operator should be familiar with DOS on an PC. The following steps are to be followed to execute the program:

1. Insert the program disk in the dirve A.
2. With the A> prompt on the screen type METRO.
3. It will ask the user for unit 1. This input file is the data file obtined from the APC. If this file is indrive B then type B: filename. File ext.
4. If there is any error, the program will inform the user that it was trying to read the data in the particular log.
5. If there was an error, then an error list is displayed on the screen.
6. Then the user is asked if he wants to change any records on the error list. If yes, then the correct values are fed into the data.
7. If no errors were found, then the total number of records in the input file is displayed and the user is asked whether he wishes to continue. If yes, then go to step 8.
8. Next, the number of days the data is present is displayed. The user is asked whether he wants to inspect the data. If yes, then a list of the type of logs, time (in real time), distance and the ons and offs are displayed on the screen. If no, it goes to to step 9.
9. The user is asked to input the route number of the bus from which the data was collected. The route names and numbers are displayed on the screen.

10. Entering the route number, the user is asked whether this route was the only assignment the bus had on that day. If no, then the user should input the second route number for the day.
11. If the answer is yes in step 10, it asks for unit 11. Unit 11 is the external file containing the distances between the stops. (For route 1 the filename is DISTAN1.DAT).
12. Then the number of stops in the route is entered. For better matching of the distance, the signboard log is also considered as a bus-stop log, hence the user should enter one stop more than its usual number. For example, in the Cove Road-Prospect Hills route there are 209 official stops, but the number of stops entered is 210. Also in the external distance file for practical purposes the distance between the signboard and the first stop is taken as 50 feet. Therefore it is suggested that for all the other route distance files the same principle should be followed.
13. If there was a second route, the user is asked to enter the number of stops in this route also.
14. Next enter the external file for the scheduled times. The schedule file for Cove Road-Prospect Hills route is SCHED1.DAT. If the schedule time on a route is different then the schedule file should contain these scheduled times. After this is done, the matching process begins and now it is ready to print the output.
15. Switching on the printer, the user is asked whether he needs to print a sample report. If yes, then press the CTRL key and PRTSC key together.

16. The program then asks for unit 6: Unit 6 is the file where the output is stored. Any name could be given to the output. For the route 1, ROUTE 1.OUT is given as the name.
17. It then asks for unit 7. This output is the raw data output with distance in feet and time in seconds. The filename for this unit is given as ROUTE1.DAT for the first route.

This file will be used for developing the Section 15 Report. Time Point Trip File and the three plots as mentioned in the earlier chapters. Thus the Vehicle Profile File (Unit 6) is obtained in this way. With the Route1.Dat file, it is now possible to obtain the above reports.

7.5 STEPS TO OBTAIN THE SECTION 15 REPORT.

The Section 15 Report program can be executed after the above output is obtained. The Route1.Dat is the data file, i.e, the input file to the Section 15 Report program.

7.5.1 Procedure

1. Insert the Section 15 program diskette in drive A.
2. If two drive system then place the diskette containing the input to Section 15 program in drive B, else the input should be in the C drive.
3. Type SECT15 and hit RETURN.

4. It will ask for Unit 5. This file is the output of the APC program. As mentioned earlier the file name is Route1.Dat. This output is actually the input to SECT15.
5. Displaying the name of the route, the user is required to input the number of stops on that route. As mentioned earlier, for practical purposes, the number of stops should be one more than the actual.
6. After calculating most of the variables of the report, the Section 15 report. Name it as SECT15.OUT. It asks again for Unit 5. This time the Unit 5 is the file of the next days data or any other days data. This procedure is followed since the data is being retrieved and processed for each day. Selecting 'yes', the user should input the filename for processing and the program again starts from Step 4. If there is a no other file for processing, go to Step 7.
7. It will now ask for Unit 6. This file is the output, i.e., the Section 15 report. Name it as SECT15.OUT.
8. Then it asks for total number of trips during the AM Peak. This number is for the fiscal year the system is in operation. Enter the number of trips for AM Peak.
9. Enter the number of trips for Midday.
10. Enter the number of trips for PM Peak.
11. Similarly enter the number of trips other than during the above periods.
12. Similarly enter the total number of Saturdays that service was operated.
13. Lastly, enter the fiscal year end, i.e., the month, day and year.

14. The output is now displayed on the screen which is similar to the one required by UMTA.

7.6 STEPS TO OBTAIN THE TIME POINT TRIP FILE.

Just like the Section 15 Report, this report is also obtained from the raw data output of the APC main program.

7.6.1 Procedure

1. Insert the Time Point Trip file diskette in drive A.
2. If two drive system then place the diskette containing the input to the program in drive B, else the input should be in the C drive.
3. Type TRIP and hit RETURN.
4. It will ask for Unit 5. This file is the output of the APC program. As mentioned earlier the file name was Route1.Dat. This is the input to obtain the Time Point Trip File.
5. Next the user is asked for Unit 6. This is the output file where the report is stored. The name Trip.Out is preferred.
6. Displaying the name of the route the user is required to enter the number of stops on that route. As mentioned earlier, the stops should be one more than the actual.
7. The statistics at the time points will be displayed continuously with the maximum load being displayed after each loop.

8. After the calculations for the input file, it asks if there is another file to be processed. If yes, then it will ask for Unit 5 again. Continue from Step 4 with the new filename.
9. The final output will be the Trip.Out which contains the Time Point Trip file.

7.7 STEPS TO OBTAIN THE DIFFERENT PLOTS

Actually in this section there are two stages. First is the execution of the fortran program, PLOTS, to obtain the input for the Symphony program, and then the execution of the Symphony macros to obtain the three plots.

7.7.1 Procedure for the frist part

This program contains three parts. Each part is useful for each of the three plots. The first part is to give out the ons at each stop that the bus arrived. The second part is the load at each stop the bus arrived. The third part gives out the load at each of the time points.

For this first two parts the input files are the same, where as for the third part the input file is the Trip2.Out obtained from the Time Trip file program. The steps are as follows:

1. The first two steps are similar to the ones explained earlier.
2. Type PLOTS and hit RETURN.

3. It will ask for Unit 1. This is the same output file obtained from the APC program.
4. It will next ask the user for the number of stops in the particular route, displaying the name of the route. Again the number of stops should be one more than the actual, for reasons mentioned earlier. This was for the Route Demand Plot.
5. Next processing for the Load Plot begins. It will ask for Unit 4. Unit 1 and Unit 4 are the same, hence the user should write the same filename.
6. Next it will ask for Unit 5. This input file is the output obtained from the Time Point Trip file program. This unit is the Trip1.Out obtained from the aforesaid program.
7. The user is then asked if there are any more data to be processed from another file, then he has to follow the steps again from Step 3.
8. If there is no more data or data files, the program will ask for Unit 2. This is the output file of this program. Preferably name it as PLOTS.OUT.

It is now ready to obtain the plots from this output.

7.7.2 Steps to obtain the Plots

Before starting the procedure, the user is assumed to be quite familiar to the procedure of initiating Symphony. If not, he is requested to refer the Symphony manual before initializing the procedure.

After entering the Symphony spreadsheet environment, the following steps are to be followed.

1. First specify the drive from where the output has to be retrieved. Hit the {Services} key (F9). Move the pointer to Directory and hit {RETURN}. Specify the drive.
2. Hit the {SERVICES} key (F9) again.
3. Move the pointer to file and hit return.
4. Move the pointer to Retrieve.
5. The file list in that drive is displayed. Move the pointer to the file PLOTS.WR1 and hit return. (This file is to retrieve first since the macro is stored in this file).
6. To import the Plots.Out file, hit the {Services} key again.
7. Move pointer to Import and hit return.
8. Move the pointer to structured and hit return.
9. Again, the file list is displayed. Either move the pointer to the file name or type the file name (PLOTS.OUT) and hit return.
10. The spreadsheet is now ready to execute the Macro. Hit the {USER} (F7) key. The words user appear at the bottom right hand corner of the screen.
11. Type PLOT (the name of the macro) to execute the macro. The name appears in place of user and hit return.
12. The macro execution starts. It will ask whether you want a few plots. Selecting 'No' stops the macro execution. If 'Yes', got to Step 13.
13. It will ask whether you have all the data here. Selecting 'Yes', asks you to import the data, and then hit return to quit.

14. Selecting 'No' in Step 13, it asks whether you want the Route Demand Plot. If yes, continue, else go to Step 25.
15. It will ask for X range. By pointing indicate the range, (the X axis is the time of the day, i.e., hours), and hit return.
16. Next it will ask for the A range. Indicate the passenger ons and hit return.
17. Type the lower limit as 6, and upper limit 24. These are the numbers along the x-axis for the time of day.
18. It will ask for the y-scale width; enter 4 and hit return.
19. Enter the first title -- Roanoke Valley Metro, hit return.
20. The second title is the name of the graph -- Route Demand Plot, hit return.
21. Type x-axis title -- Time of Day, hit return.
22. Type the y-axis title as -- Passengers Boarded, and hit return.
23. The graph is displayed, hit return to continue the macro again.
24. If you want to save this plot, it will ask you for a filename. Type the name and hit return.
25. It will now ask you whether you want the Route Demand Bar Plot. If yes, continue, else go to Step 27.
26. The steps are similar to the steps from 15 to 24, except for some minor changes.
27. It will now ask the user whether he wants the Passenger Load Plot. If yes, continue, else go to Step 29.
28. The procedure is same from Step 15 to Step 24, except that the X-range and A-range are the Arrival and Passenager load columns respectively.

29. Next it will ask whether you need the Load Profile Plot. If yes, go to Step 30, else go to Step 31.
30. If the answer is 'yes' in Step 29, then once again the procedure is the same as from Step 15 to Step 24 except that the X-range and A-range will be the stop numbers (time points) and the load at the time points, respectively.
31. Next it will ask the user whether he needs the Load Profile Bar Plot. If no, go to Step 33.
32. The steps are same as in Step 30, including the X-range and A-range.
33. The user is then asked if he needs any of the other three plots. Select yes if you have missed any plots, else select no.
34. Next it asks the user whether he needs to print any plots.
35. If yes in Step 34, please exit after saving this file.

To print a graph, the user has to go to the Printgraph environment. Then to print the graph, it is advised to follow the manual for the configuration and settings.

8.0 CONCLUSIONS AND RECOMMENDATIONS

Information on transit ridership and bus performance is essential for eficient transit management Automatic Passenger Counting systems monitor the progress of a particular vehicle- it's present position, the number of passengers on board and the current time- and makes this readily available for processing into valuable data for the schedulers, planners and managers.

Transit decision-making can be more timely because data from an automated system will generally be available soomner than data from a manual data collection system. Thus, the data turnaround time is less than the manual collection. One of the best points in favor of the APC system, is that it will permit a continuous data collection effort that is definitely more time effective, and therefore, more cost effective.

One most important conclusion of this thesis is that several computerized reports could be obtained from the APC. These reports could assist managers and schedulers in effective decision making. The different plots would also assist the managers to organize or schedule activities for planning and control purposes. Graphical display of data that traditionally have been presented in tables may allow managers to compare data values, observe patterns and detect trends more easily. The use of computer graphics for reporting and decision making presents new questions to researchers.

Another point that is brought forward here, is that, since there is no signpost in the system, the sign board log should be created everytime,

every day, on every route, at the same stop. If there is any change in this record, the distance matching process will be mis-matched giving out wrong stop statistics. To avoid this the bus operators should be informed well in advance to change the signboard at the same stop everytime.

Whenever there were multiple entries, the sensors used to miss at least one passenger. Hence it is recommended that certain measure should be taken to avoid this problem.

As observed in the ten loops, the error in 'ons' was found to be only 6%, and for 'offs' it was about 5%. These low figures indicate the reliability and acuracy of the APC. Even the distance measured by the APC had a low error, with an average error over the ten loops being 0.25 miles (1350 feet). Also the probability of an error occuring at at each stop on the route was found to be 1.6% and 1.1% for 'ons' and 'offs' respectively.

It is strongly recommended here that more reliable odometer readings be obtained so that proper referencing and matching of the data could be achieved. A well caliberated odometer, which does not break down frequently, should be developed, which will enhance the referencing process of the software.

Also, recommended is the usage of signposts instead of the signboard record. This signboard record is more dependent on the operator, thus making it vulnerable to get wrong data. There should not be any human involvment, hence it is suggested to have a signpost along the route. More signposts will of course improve the reliability and accuracy of the system.

To obtain section 15 report, it is suggested to have more APC's, for elaborating the section 15 requirements, as is being actually done in other applications.

Finally, it is suggested that the reports and results of this project be made available to the other transit companies, irrespective of their size, that are contemplating the implementation of an automated data collection system. This will help the transit companies in offering a safer, better and more reliable service to the general public at large, thus increasing the popularity of the mass transit.

REFERENCES

1. Bonsall, John A. OC Transpo Data Collection System, Ottawa-Carleton Regional Transit Commission, Ottawa, Ontario, Sept. 1978. 24pps.

2. Diebel, L.E. and Zumwalt, B. A Modular Approach to On-Board Automatic Data Collection Systems. NCTRP Report 9, Transportation Research Board, Washington, D.C., Dec. 1984, 123 pps.

3. Geehan T. and N.V. McEachern. APC Uniform Functional Requirement Definition. Prepared By CUTA. Nov. 1983. 66 pps.

4. Metro Transit System. APC Demonstration and Development Program. Presented at the ATA Conference, Clorado, October 1983. 12 pps.

5. Poirer, P.J. and Hobbs, V.J. Synopsis of the Working Group Meeting on Automated Passenger Counting Systems. Held at Cambridge, MA. 30 pps.

6. Vozzolo, D. and Attanucci, J. An Assessment of Automatic Passenger Counters. UMTA, Oct. 1982, 12 pps.

7. Microsoft Fortran Compiler Reference Manual. Microsoft Corporation, 1984. 400 pps.

8. Merchan, Micheal. Fortran 77 Languageand Style., Wadsworth Pub., Belmont, CA. 1981. 445 pps

9. Vozzolo, D. and Attanucci, J. An Operational Effectiveness Accuracy and Costs of Automatic Passenger Counters. TRB 947, Wash. D.C. 1983 56 pps.

10. Urban Transportation Associates. APC Catalogue of Software Capabilities. Ohio, 1984. 100 pps.

11. Attanucci, John, et al. Bus Transit Monitoring Manual, Volumes 1 and 2. ATE Co., Aug 1981, 152 pps.

12. Urban Transportation Associates, Catalogue of Software Capability, Cincinnati, Ohio.

13. U.S. DOT, Urban Mass Transportation Industry Uniform System of Accounts and Records and Reporting System. June 1985.

14. Gerardine DeSantis, Computer Graphicsas Decision Aids. Decision Sciences, Vol. 15, 1984.

15. Keen Peter G.W. and Scott-Morton Micheal S., Decision Support System, 1978, Pub. Addison-Wesley, Reading, MA.

16. Lotus Symphony Manual, Lotus Development Corporation, 1985.

17. Edgar D.L. Izeppi, An Automatic Passenger Counting System at a Small Transit Property, 1985.

APPENDIX A. APC COMPUTER PROGRAMS

```
C*************************************************************************
C  PROGRAM NAME: AUTOMATIC PASSENGER COUNTER
C*************************************************************************
C
C  Written by: EDGAR DE LEON
C  Date      : FEBRUARY 18, 1985.
C
C  Purpose: READS AUTOMATIC PASSENGER COUNTER DATA AND STORES IT AFTER
C           MAKING CERTAIN VALIDITY CHECKS
C
C     THIS PROGRAM READS A SET OF INPUT LOGS TAKEN FROM THE ON-BUS
C     AUTOMATIC PASSENGER COUNTER. IT CHECKS EACH LOG FOR VALID VALUES.
C     ACCEPTABLE VALUES ARE THEN COPIED TO A NEW FILE. UNACCEPTABLE
C     VALUES ARE DIAGNOSED ON A LISTING.
C
C     TO BE VALID EACH VALUE MUST BE POSITIVE AND CONTAIN A VALUE THAT
C     IS GREATER THAN OR EQUAL TO XMIN AND LESS THAN OR EQUAL TO XMAX.
C
C     LOG    = RECORD NUMBER OF INPUT LOGS
C     ERRCNT = NUMBER OF ERRORS
C     ERROR  = ARRAY OF ERROR POINTERS
C     NGOOD  = NUMBER OF GOOD DATA VALUES
C     NDAT   = NUMBER OF DATA VALUES PER INPUT LOG
C     STAT   = INPUT-OUTPUT STATUS
C
      COMMON /LOGNO/ LOG,LLOG
      COMMON /ERRUR/ ERRCNT, ERROR
      INTEGER LOG, ERRCNT, ERROR(25),LLOG
      INTEGER NGOOD, NDAT, STAT
      PARAMETER (NDAT = 5)
C
C
C     TYPE(I) = INPUT TYPE OF EVENT I, AS A INTEGER NUMBER
C     TIME(I) = INPUT TIME OF EVENT I, AS A INTEGER NUMBER
C     DIST(I) = INPUT DISTANCE OF EVENT I, AS A INTEGER NUMBER
C     ONS(I)  = INPUT BOARDINGS OF EVENT I, AS A INTEGER NUMBER
C     OFFS(I) = INPUT ALIGHTINGS OF EVENT I, AS A INTEGER NUMBER
C             = IT IS ALSO ON EVENTS THAT HAVE A SIGNPOST DETECTION THE
C               INPUT SIGNPOST NUMBER OF EVENT I, AS A INTEGER NUMBER
C
C     XMIN   = MINIMUM ACCEPTABLE INPUT VALUE (ZERO)
C     XMAX(1)  = MAXIMUM ACCEPTABLE INPUT VALUE FOR TYPE OF EVENT
C     XMAX(2)  = MAXIMUM ACCEPTABLE INPUT VALUE FOR TIME OF EFENT
C     XMAX(3)  = MAXIMUM ACCEPTABLE INPUT VALUE FOR DISTANCE OF EVENT
C     XMAX(4)  = MAXIMUM ACCEPTABLE INPUT VALUE FOR ON'S OF EVENT
C     XMAX(5)  = MAXIMUM ACCEPTABLE INPUT VALUE FOR OFF'S OF EVENT
C              = OR MAXIMUM ACCEPTABLE INPUT VALUE OF SIGNPOST I.D. #
C
```

```
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER X(NDAT), XMIN, XMAX(NDAT)
      INTEGER TYPO(1201), TIMO(1201), DISTO(1201),ONO(1201),OFFO(1201)
      INTEGER A, P
C
C     VALID  = TRUE IF INPUT RECORDS ARE VALID
C
      LOGICAL VALID
C
C     THE FOLLOWING VARIABLES ARE THE MANUALLY INPUTTED DATA:
C     HH, MM, SS, AND PRSNID IS THE FIRST TYPE 12 LOG,
C     DAY, MONTH,YEAR, AND MACHID (MACHINE NUMBER) IS THE SECOND.
C     FOR CONVENIENCE THE FOLLOWING SUBSTITUTION WAS MADE IN THE
C     NAME OF THE VARIABLES, SO THAT THEY DON'T REPEAT AND CONFUSE:
C     MES = MONTH
C     ANO = YEAR
C
      COMMON /INPUT1/ DAY, MES, ANO, WKDAY
      COMMON /INPUT2/ MACHID, HH, MM, SS, PRSNID
      INTEGER DAY, MES, ANO
      INTEGER MACHID, HH, MM, SS, WKDAY
      CHARACTER PRSNID
C
C     THE NEXT COMMON BLOCK DEFINES THE DATE FOR EACH INDIVIDUAL BLOCK.
C     FOR EASE AND TO AVOID CONFUSION, THE COMMON BLOCK WAS NAMED FECHA
C     WHICH MEANS "DATES".
C
      COMMON /FECHA/ DATE, MONTH, YEAR, IDAY
      INTEGER DATE(1201), MONTH(1201), YEAR(1201), IDAY(1201)
C
      COMMON /ROUTE/ RUTENO
      COMMON /BLOCK/ BLOKNO
     • INTEGER BLOKNO(1201), RUTENO(1201)
C
      XMIN = 0
      XMAX(1) =  12
      XMAX(2) = 240
      XMAX(3) = 255
      XMAX(4) =  60
      XMAX(5) =  60
C
C     OPEN INPUT FILE
C
      OPEN (UNIT = 1, FILE = ' ', IOSTAT = STAT)
C
C     INITIALIZE COUNTERS
C
      LLOG = 0
      LOG = 0
      ERRCNT = 0
```

```
      NGOOD  = 0
      A = 2
C
C     FOR EACH INPUT LOG:
C
      DO 500 J = 1, 1201
        READ (1, * , END = 900, ERR = 97, IOSTAT = STAT)
     *  (X(I), I = 1, NDAT)
        LLOG = LLOG + 1
        VALID = .TRUE.
C
C     CHECK FOR VALIDITY
C
          DO 100 I = 1, NDAT
           IF (STAT .EQ. 0) THEN
             IF (XMIN .LE. X(I) .AND. X(I) .LE. XMAX(I))  THEN
                  TYPO(LLOG) = X(1)
                  TIMO(LLOG) = X(2)
                  DISTO(LLOG) = X(3)
                  ONO(LLOG)  = X(4)
                  OFFO(LLOG) = X(5)
                  GO TO 100
             ENDIF
               VALID = .FALSE.
           ELSE
               GO TO 97
           ENDIF
100       CONTINUE
        IF (VALID) THEN
          NGOOD = NGOOD + 1
        ELSE
          ERRCNT = ERRCNT + 1
          ERROR(ERRCNT) = LLOG
        ENDIF
      IF (TYPO(LLOG) .EQ. 7 .OR. TYPO(LLOG) .EQ. 8) THEN
          XMIN = 0
          XMAX(1) =  12
          XMAX(2) =  24
          XMAX(3) =  60
          XMAX(4) = 100
          XMAX(5) = 100
      ENDIF
C
      IF (TYPO(LLOG) .EQ. 12) A = A - 1
      IF (A .EQ. 0) GO TO 900
500   CONTINUE
C
C     OPEN OUTPUT FILES
C
      OPEN (UNIT = 6, FILE = ' ', STATUS = 'NEW')
C
```

```
      OPEN (UNIT = 7, FILE = ' ', STATUS = 'NEW')
C
C     AN ERROR OCURRED DURING THE INPUT
C
97    WRITE (*, 110) STAT, LLOG
110   FORMAT(1X, 'ERROR NUMBER', I4, 2X, 'TRYING TO READ THE DATA',
     * 'IN LOG', I4)
      GO TO 600
C
C     AT THE END OF THE INPUT THE NUMBER OF DATA RECORDS, ERRORS, AND
C     OTHER SYSTEM STATISTICS DISPLAYED ON THE SCREEN SO THAT THE
C     OPERATOR CAN CHECK THE INPUT PROCESSING.
C
900   WRITE (*,115) LLOG, ERRCNT, NGOOD
115   FORMAT (3X, 'THERE WERE A TOTAL OF', I5, 2X, 'RECORDS READ WITH',
     * I4, 2X, 'DATA ERRORS,',  /, 3X, 'AND A TOTAL OF', I5, 2X,
     * 'ACCEPTABLE DATA RECORDS')
      IF (ERRCNT .GT. 0) THEN
         CALL ERRORS
         WRITE (*, *)    '   DO YOU WISH TO CONTINUE ? YES = 1, NO = 0'
         READ (*, *) M
         IF (M .EQ. 0) GO TO 600
         GO TO 910
      ENDIF
      WRITE (*, *)       '   DO YOU WISH TO CONTINUE ? YES = 1, NO = 0'
      READ (*, *) M
      IF (M .EQ. 0) GO TO 600
C
C  IN CASE THE MANUALLY INPUT DATA IS MISSING IN THE DATA FILE, PLEASE
C  INPUT IT SO THAT THE ADEQUATE PROCESSING CAN BE DONE
C
      DO 55 P = 1,LLOG
       IF (TYPO(P) .EQ. 8 .OR. TYPO(P) .EQ. 9 .OR. TYPO(P) .EQ. 10) THEN
             GO TO 55
       ELSE
             LOG = LOG + 1
                 TYPE(LOG) = TYPO(P)
                 TIME(LOG) = TIMO(P)
                 DIST(LOG) = DISTO(P)
                 ONS(LOG) = ONO(P)
                 OFFS(LOG) = OFFO(P)
       ENDIF
C       WRITE (30,*) TYPE(LOG), TIME(LOG),DIST(LOG),ONS(LOG),OFFS(LOG)
55    CONTINUE
C
910   IF (A .NE. 0) THEN
       CALL TIMEIN
       GO TO 950
      ENDIF
C
C  IF THE MANUALLY INPUT DATA WAS PRESENT, THEN START ASSIGNING EACH
```

```
C  OF THIS DATA TO ITS RESPECTIVE VARIABLE NAMES SO THAT LATER THE
C  BACKTAGGING OF ALL THE DATA LOGS IS DONE FROM THIS REFERENCE POINT.
C
      CALL REFRNS
950   TIME(LOG - 1) = HH*3600 + MM*60 + SS
C
C  WITH THE REAL TIME AND DATE, START BACKTAGGING ALL OF THE DATA
C  RECORDS WITH THE BAKTAG SUBROUTINE. THIS WILL ENABLE THE PROCESSING
C  OF THE DATA TO BE DONE FOR EACH DAY, LATER ON.
C
      CALL BAKTAG
C
C  WITH THE REAL TIME ALL OF THE DATA LOG RECORDS CAN BE DIVIDED BY THE
C  DATE IN WHICH THEY WERE MADE. IT IS POSSIBLE TO INPECT THE DATA BY
C  DATE, IF NECESSARY. THIS WILL BE DONE BY THE VICHEK SUBROUTINE.
C
      CALL SEGMNT
C
600   STOP
      END
C
C     SUBROUTINE ERRORS
C
      SUBROUTINE ERRORS
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /ERRUR/ ERRCNT, ERROR
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER ERRCNT, ERROR(25)
      INTEGER TYPO(1201), TIMO(1201),DISTO(1201), ONO(1201), OFFO(1201)
C
      WRITE (*, 300)
300   FORMAT ('1', /, 25X, 'ERROR LIST',///, 2X, 'LOG', 5X, 'TYPE', 2X,
     * 'TIME', 5X, 'DIST', 6X, 'ON', 5X, 'OFF', /)
C
      DO 100 I = 1, ERRCNT
         J = ERROR(I)
         WRITE (*, 400) J, TYPO(J), TIMO(J), DISTO(J), ONO(J), OFFO(J)
400      FORMAT (I5, 5I8, /)
100   CONTINUE
      IF (ERRCNT .GT. 10) THEN
         WRITE (*, *)    '   TOO MANY ERRORS, DO YOU STILL WANT TO CONTIN
     *UE? YES = 1, NO = 0'
         READ (*, *) II
         IF (II .EQ. 0) GO TO 500
      ENDIF
      WRITE (*, *)       '   DO YOU WANT TO CHANGE THE VALUES OF THE RECO
     *RDS LISTED ABOVE? YES = 1,NO = 0'
      READ (*, *) JJ
      IF (JJ .EQ. 0) GO TO 500
C
      DO 200 K = 1, ERRCNT
```

```
          L = ERROR(K)
          WRITE (*, 310)
310       FORMAT ('1', 2X, 'REC', 5X, 'TYPE', 2X, 'TIME', 5X, 'DIST',
     * 6X, 'ON', 5X, 'OFF', /)
          WRITE (*, 410) L, TYPO(L), TIMO(L), DISTO(L), ONO(L), OFFO(L)
410       FORMAT (I5, 5I8)
          WRITE (*, *)    '   DO YOU WANT TO CHANGE THE DATA ON THIS RECOR
     *D? YES = 1, NO = 0'
          READ (*, *) KK
          IF (KK .EQ. 0) GO TO 200
          WRITE (*, *)    '   PLEASE INPUT THE CORRECT INFORMATION.
     *                                    TYPE = '
          READ (*, *) TYPO(L)
          WRITE (*, *)  '   TIME =  '
          READ (*, *) TIMO(L)
          WRITE (*, *)  '   DIST =  '
          READ (*, *) DISTO(L)
          WRITE (*, *)  '   ONS =   '
          READ (*, *) ONO(L)
          WRITE (*, *)  '   OFFS =  '
          READ (*, *) OFFO(L)
200     CONTINUE
500     RETURN
        END
C
        SUBROUTINE TIMEIN
        COMMON /INPUT1/ DAY, MES, ANO, WKDAY
        COMMON /INPUT2/ MACHID, HH, MM, SS, PRSNID
        COMMON /LOGNO/ LOG
        INTEGER DAY, MES, ANO, IDAY
        INTEGER MACHID, HH, MM, SS, LOG
        CHARACTER PRSNID
C
        WRITE (*, 120)
120     FORMAT (3X, 'THE LAST RECORDS READ WERE NOT THE MANUALLY INPUT ',
       * 'DATA', /, 3X, 'PLEASE ENTER THE APPROPIATE TIME AND DATE WHEN'
       * 'THE DATA WAS DUMPED', /, 3X, 'HIT RETURN AFTER ENTERING THE'
       * 'NUMBER')
        WRITE (*, *)     '   ENTER THE TIME; HOURS = '
        READ (*, *) HH
        WRITE (*, *)     '   MINUTES = '
        READ (*, *) MM
        WRITE (*, *)     '   SECONDS = '
        READ (*, *) SS
        WRITE (*, *)     '   ENTER YOUR FIRST INITIAL (DAY #) = '
        READ (*, *) PRSNID
        WRITE (*, *)     '   NOW ENTER THE DATE; DAY = '
        READ (*, *) DAY
        WRITE (*, *)     '   MONTH = '
        READ (*, *) MES
        WRITE (*, *)     '   YEAR = '
```

```
      READ (*, *) ANO
      WRITE (*, *)     '    ENTER THE BUS I.D. NUMBER = '
      READ (*, *) MACHID
      LOG = LOG + 2
      RETURN
      END
C
      SUBROUTINE REFRNS
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /INPUT1/ DAY, MES, ANO, WKDAY
      COMMON /INPUT2/ MACHID, HH, MM, SS, PRSNID
      COMMON /LOGNO/ LOG
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER DAY, MES, ANO
      INTEGER MACHID, HH, MM, SS, LOG, WKDAY
      CHARACTER PRSNID
C
      MES = TIME(LOG)
      DAY = DIST(LOG)
      ANO = ONS(LOG)
      WKDAY = OFFS(LOG)
      HH = TIME(LOG - 1)
      MM = DIST(LOG - 1)
      SS = ONS(LOG - 1)
      PRSNID = OFFS(LOG - 1)
      RETURN
      END
C
      SUBROUTINE BAKTAG
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /FECHA/ DATE, MONTH, YEAR, IDAY
      COMMON /LOGNO/ LOG
      COMMON /INPUT1/ DAY, MES, ANO, WKDAY
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER DATE(1201), MONTH(1201), YEAR(1201), IDAY(1201)
      INTEGER LOG, DAY, MES, ANO, WKDAY
      INTEGER TEMP, MINUS
C
      TEMP = TIME(LOG - 2)
      TIME(LOG - 2) = TIME(LOG - 1)
      DATE(LOG - 2) = DAY
      MONTH(LOG - 2) = MES
      YEAR(LOG - 2) = ANO
      IDAY(LOG - 2) = WKDAY
C
      DO 100 I = LOG - 2, 2, -1
         IF (TYPE(I) .EQ. 2) THEN
           MINUS = 240
         ELSE
           MINUS = TEMP
         ENDIF
```

```
          TEMP = TIME(I - 1)
          TIME(I - 1) = TIME(I) - MINUS*15
          IF (TIME(I - 1) .LE. 0 ) THEN
            TIME(I - 1) = 86400 + TIME(I - 1)
            DAY = DAY - 1
            WKDAY = WKDAY - 1
          ENDIF
C
         IF (WKDAY .EQ. 0) THEN
             WKDAY = 6
         ENDIF
C
          IF (DAY .EQ. 0) THEN
            GO TO (41, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30) MES
C
41          DAY = 31
            ANO = ANO - 1
            MES = 12
            GO TO 199
31          DAY = 31
            GO TO 99
30          DAY = 30
            GO TO 99
28          DAY = 28
            IF ( MOD (YEAR(I) , 4) .EQ. 0) DAY = 29
99          MES = MES - 1
          ENDIF
199       DATE(I - 1) = DAY
          MONTH(I - 1) = MES
          YEAR(I - 1) = ANO
          IDAY(I - 1) = WKDAY
100    CONTINUE
       RETURN
       END
C
C  THIS IS SUBROUTINE SEGMNT. IT SEGMENTS THE DATA RECORDS BY DAY.
C
       SUBROUTINE SEGMNT
       COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
       COMMON /FECHA/ DATE, MONTH, YEAR
       COMMON /ROUTE/ RUTENO
       COMMON /BLOCK/ BLOKNO
       COMMON /LOGNO/ LOG
       COMMON /INPUT2/ MACHID, HH, MM, SS, PRSNID
       INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
       INTEGER DATE(1201), MONTH(1201), YEAR(1201)
       INTEGER RUTENO(1201), BLOKNO(1201)
       INTEGER LOG
       INTEGER MACHID, HH, MM, SS
       CHARACTER PRSNID
C
```

```
      COMMON /CONTEO/ COUNT, MARK
      INTEGER COUNT, MARK(20)
C
C  INITIALIZE COUNTERS
C
      COUNT = 1
      MARK(COUNT) = 0
      DO 200 I = 1, LOG - 3
        IF (DATE(I) .NE. DATE(I + 1)) THEN
           COUNT = COUNT + 1
           MARK(COUNT) = I
        ENDIF
200   CONTINUE
      COUNT = COUNT + 1
      MARK(COUNT) = LOG - 2
C
C  OPTIONAL VISUAL CHECK OF THE DATA RECORDS BY DAY
C
      WRITE (*, 100) COUNT - 1
100   FORMAT ('1', 3X, 'THE DATA TO BE PROCESSED INCLUDES DATA FOR ',
     * I5, 2X, 'DAYS', /)
      WRITE (*, *)       '    DO YOU WANT TO VISUALLY INSPECT THE DATA? YE
     *S = 1, NO = 0'
      READ (*, *) MM
      IF (MM .EQ. 0) GO TO 500
      CALL VICHEK
500   CALL BLOCK1
      RETURN
      END
C
C  THIS SUBROUTINE IS IN CHARGE OF MAKING ALL OF THE DATA RECORDS
C  AVAILABLE FOR A SCREEN PRE-VIEWING. IT IS PARTICULARLY VERY USEFUL
C  WHEN THERE HAVE BEEN CORRECTIONS MADE AT THE BEGINING OF THE PROCESS.
C
      SUBROUTINE VICHEK
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /FECHA/ DATE, MONTH, YEAR
      COMMON /CONTEO/ COUNT, MARK
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER DATE(1201), MONTH(1201), YEAR(1201)
      INTEGER COUNT, MARK(20)
      INTEGER LINENO, JJ
C
      JJ = 1
C
      DO 100 J = 1, COUNT - 1
        M = MARK(J) + 1
        N = MARK(J + 1)
        LINENO = 0
        DO 200 I = M, N
          IF (LINENO .EQ. 0) THEN
```

```
            WRITE (*, 300)
300         FORMAT ('1', 2X, 'LOG', 5X, 'TYPE', 4X, 'TIME', 6X, 'DIST',
     *       6X, 'ON', 5X, 'OFF', 6X, 'DATE', /)
          ENDIF
          IH = TIME(I)/3600
          IM = (TIME(I) - IH*3600)/60
          IS = TIME(I) - IH*3600 - IM*60
          WRITE (*, 320) JJ, TYPE(I),IH, IM, IS, DIST(I),
     *      ONS(I), OFFS(I), MONTH(I), DATE(I), YEAR(I)
320       FORMAT (I5, I8, 4X, I2, ':', I2, ':', I2, 3I8, 3X, I2,
     *      '/', I2, '/', I2)
          LINENO = LINENO + 1
          JJ = JJ + 1
          IF (I .EQ. N) LINENO = 20
          IF (LINENO .GE. 20) THEN
            LINENO = 0
            WRITE (*, *)'   WANT TO CONTINUE INSPECTION? YES = 1, NO = 0
     *'
            READ (*, *) II
            IF (II .EQ. 0) GO TO 500
          ENDIF
200     CONTINUE
100   CONTINUE
500   RETURN
      END
C
C     SUBROUTINE BLOCK1 INPUTS THE ROUTE AND BLOCK NUMBERS FOR THE DATA
C
      SUBROUTINE BLOCK1
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /ROUTE/ RUTENO
      COMMON /BLOCK/ BLOKNO
      COMMON /FECHA/ DATE, MONTH, YEAR
      COMMON /CONTEO/ COUNT, MARK
      COMMON /SIZE/ LIMITK, LIMITJ
      COMMON /INPUT2/ MACHID, HH, MM, SS, PRSNID
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER RUTENO(1201), BLOKNO(1201)
      INTEGER DATE(1201), MONTH(1201), YEAR(1201)
      INTEGER COUNT, MARK(20)
      INTEGER LIMITK, LIMITJ
      INTEGER ROUTE1, ROUTE2, BLOCK1, BLOCK2
      INTEGER MACHID, HH, MM, SS
      CHARACTER PRSNID
C
C     THE FOLLOWING COMMON BLOCKS ARE USED BY THE OPEN SUBROUTINE
C
      COMMON /DISTA1/ FROM1, TO1, DIST1, NAME1
      COMMON /DISTA2/ FROM2, TO2, DIST2, NAME2
      COMMON /DIAS/ M, N
      INTEGER FROM1(300), TO1(300), DIST1(300)
```

```
      INTEGER FROM2(300), TO2(300), DIST2(300)
      CHARACTER*60 NAME1(300), NAME2(300)
      INTEGER M, N
C
      DO 100 J = 1, COUNT - 1
         M = MARK(J) + 1
         N = MARK(J + 1)
         IF ( (N-M) .LT. 24) GO TO 100
         WRITE (*, 300) MONTH(M), DATE(M), YEAR(M)
300      FORMAT ('1', 3X, 'ON WHICH ROUTE WAS THE BUS WORKING ON THE',
     *           ' FOLLOWING DAY ?', /, 3X, 'DATE :  ',
     *            I2, '/', I2, '/', I2, /)
         WRITE (*, 401)
401      FORMAT( 4X, 'COVE ROAD/PROSPECT HILLS          = 1   ', /,
     *           4X, 'LINCOLN TERRACE/SALEM-SHENANDOAH = 2   ', /,
     *           4X, 'RIVERDALE/RUGBY                  = 3   ', /,
     *           4X, 'KENWOOD/VINTON-WISE AVE          = 4   ', /,
     *           4X, 'MELROSE/VIRGINIA HEIGHTS         = 5   ', /,
     *           4X, 'HUNTINGTON COURT/WASENA          = 6   ', /,
     *           4X, 'RALEIGH COURT/VILLA HEIGHTS      = 7   ', /,
     *           4X, 'COLONIAL HEIGHTS/WILLIAMSON ROAD = 8   ', /,
     *           4X, 'GRANDIN COURT                    = 9   ', /,
     *           4X, 'EAST ORANGE AVENUE               = 10  ', /)
         WRITE (*, *)   '   ROUTE NUMBER =  '
         READ (*, *)  ROUTE1
         WRITE (*, *)   '   IS THIS ROUTE THE ONLY ASSIGNMENT THE BUS HA
     *D ON THAT DAY ? YES = 1, NO = 0'
         READ (*, *) IJ
         IF (IJ .EQ. 1) GO TO 350
         WRITE (*, 400)
400      FORMAT( 4X, 'COVE ROAD/PROSPECT HILLS          = 1   ', /,
     *           4X, 'LINCOLN TERRACE/SALEM-SHENANDOAH = 2   ', /,
     *           4X, 'RIVERDALE/RUGBY                  = 3   ', /,
     *           4X, 'KENWOOD/VINTON-WISE AVE          = 4   ', /,
     *           4X, 'MELROSE/VIRGINIA HEIGHTS         = 5   ', /,
     *           4X, 'HUNTINGTON COURT/WASENA          = 6   ', /,
     *           4X, 'RALEIGH COURT/VILLA HEIGHTS      = 7   ', /,
     *           4X, 'COLONIAL HEIGHTS/WILLIAMSON ROAD = 8   ', /,
     *           4X, 'GRANDIN COURT                    = 9   ', /,
     *           4X, 'EAST ORANGE AVENUE               = 10  ', /)
         WRITE (*, *)   '   WHAT IS THE SECOND ROUTE ASSIGNMENT OF THE DA
     *Y?  ROUTE =  '
         READ (*, *) ROUTE2
         WRITE (*, *)   '   PLEASE INPUT THE TIME WHEN THE BUS STARTED O
     *N ANOTHER ROUTE, INPUT HOURS='
         READ (*, *) IHH1
         WRITE (*, *)   '   MINUTES = '
         READ (*, *) IMM1
         ITIME1 = IHH1*3600 + IMM1*60
350      WRITE (*, *)   '   WHAT IS THE BLOCK NUMBER ASSIGNED TO THE BUS
     * ON THIS DAY ?  BLOCK NUMBER = '
```

```
          READ (*, *) BLOCK1
          WRITE (*, *)    '    IS THIS BLOCK THE ONLY ASSIGNMENT THE BUS HA
        *D DURING THE DAY? YES = 1,NO = 0'
          READ (*, *) JI
          IF (JI .EQ. 1) GO TO 450
          WRITE (*, *)    '    WHAT IS THE SECOND BLOCK ASSIGNMENT OF THE D
        *AY?  BLOCK =  '
          READ (*, *) BLOCK2
          WRITE (*, *)    '    PLEASE INPUT THE TIME WHEN THE BUS STARTED O
        *N ANOTHER BLOCK, INPUT HOURS='
          READ (*, *) IHH2
          WRITE (*, *)    '    MINUTES = '
          READ (*, *) IMM2
          ITIME2 = IHH2*3600 + IMM2*60
          WRITE (*, *)    '    IS THERE A THIRD BLOCK ASSIGNMENT OF THE BUS
        * DURING THE DAY? YES = 1,NO = 0'
          READ (*, *) JI
          IF (JI .EQ. 1) GO TO 450
          WRITE (*, *)    '    WHAT IS THE THIRD BLOCK ASSIGNMENT OF THE DA
        *Y?  BLOCK =  '
          READ (*, *) BLOCK3
          WRITE (*, *)    '    PLEASE INPUT THE TIME WHEN THE BUS STARTED O
        *N THE THIRD BLOCK, INPUT HOURS='
          READ (*, *) IHH3
          WRITE (*, *)    '    MINUTES = '
          READ (*, *) IMM3
          ITIME3 = IHH3*3600 + IMM3*60
C
450       DO 200 I = M, N
            IF (ITIME1 .NE. 0 ) THEN
               IF (TIME(I) .GE. ITIME1) ROUTE1 = ROUTE2
            ENDIF
            IF (ITIME2 .NE. 0 ) THEN
               IF (TIME(I) .GE. ITIME2) BLOCK1 = BLOCK2
            ENDIF
            IF (ITIME3 .NE. 0 ) THEN
               IF (TIME(I) .GE. ITIME3) BLOCK1 = BLOCK3
            ENDIF
            RUTENO(I) = ROUTE1
            BLOKNO(I) = BLOCK1
200     CONTINUE
        CALL OPEN
        CALL MATCH
100     CONTINUE
        RETURN
        END
C
C       SUBROUTINE OPEN: THIS SUBROUTINE OPENS THE DATA FILES CONTAINING
C       THE STOP DATA FOR THE ROUTES COVERED BY A BUS DURING ONE DAY
C       WHICH WILL BE USED FOR REFERENCING THE DATA COLLECTED.
C
```

```
      SUBROUTINE OPEN
      COMMON /ROUTE/ RUTENO
      COMMON /DISTA1/ FROM1, TO1, DIST1, NAME1
      COMMON /DISTA2/ FROM2, TO2, DIST2, NAME2
      COMMON /DIAS/ M, N
      COMMON /SIZE/ LIMITK, LIMITJ
      INTEGER RUTENO(1201)
      INTEGER FROM1(300), TO1(300), DIST1(300)
      INTEGER FROM2(300), TO2(300), DIST2(300)
      CHARACTER*60 NAME1(300), NAME2(300)
      INTEGER M, N
      INTEGER LIMITK, LIMITJ
      INTEGER FLAG1, FLAG2, FLAG3, FLAG4
      INTEGER UNIT1, UNIT2
C
C     INITIALIZE THE FLAGS
C
      FLAG1 = 0
      FLAG2 = 0
      FLAG3 = 0
      FLAG4 = 0
      GO TO (11, 12, 12, 12, 12, 12, 12, 12, 12, 12) RUTENO(M)
11    OPEN (UNIT = 11, FILE = ' ')
      FLAG1 = 1
      GO TO 500
12    OPEN (UNIT = 12, FILE = ' ')
      FLAG2 = 1
500   DO 100 I = M, N - 1
        IF (RUTENO(I) .NE. RUTENO(I + 1)) THEN
          GO TO (1, 2, 2, 2, 2, 2, 2, 2, 2, 2) RUTENO(I + 1)
1           OPEN (UNIT = 11, FILE = ' ')
            FLAG3 = 1
            GO TO 3
2           OPEN (UNIT = 12, FILE = ' ')
            FLAG4 = 1
3       ENDIF
100   CONTINUE
      IF (FLAG1 .EQ. 1) THEN
        IF (FLAG4 .EQ. 1) GO TO 304
        GO TO 301
      ENDIF
      IF (FLAG2 .EQ. 1) THEN
        IF (FLAG3 .EQ. 1) GO TO 303
        GO TO 302
      ENDIF
301   UNIT1 = 11
      WRITE (*, *) 'ENTER THE NUMBER OF STOPS IN THE FIRST ROUTE '
      READ (*, *) LIMITK
      GO TO 300
302   UNIT1 = 12
      WRITE (*, *) 'ENTER THE NUMBER OF STOPS IN THE FIRST ROUTE '
```

```
      READ (*, *) LIMITK
      GO TO 300
303   UNIT1 = 12
      UNIT2 = 11
      WRITE (*, *) 'ENTER THE NUMBER OF STOPS IN THE FIRST ROUTE '
      READ (*, *) LIMITK
      WRITE (*, *) 'ENTER THE NUMBER OF STOPS IN THE SECOND ROUTE '
      READ (*, *) LIMITJ
      GO TO 250
304   UNIT1 = 11
      UNIT2 = 12
      WRITE (*, *) 'ENTER THE NUMBER OF STOPS IN THE FIRST ROUTE '
      READ (*, *) LIMITK
      WRITE (*, *) 'ENTER THE NUMBER OF STOPS IN THE SECOND ROUTE '
      READ (*, *) LIMITJ
C
250   READ (UNIT2, 550, END = 300)(FROM2(J), TO2(J), DIST2(J), NAME2(J),
     * J = 1, LIMITJ)
300   READ (UNIT1, 550, END = 900)(FROM1(K), TO1(K), DIST1(K), NAME1(K),
     * K = 1, LIMITK)
550   FORMAT (2I5, I8, A60)
900   RETURN
      END
C
C  SUBROUTINE MATCH: THIS SUBROUTINE IS THE ONE IN CHARGE OF STARTING
C  THE MATCHING PROCESS OF THE DATA WITH THE STOP DISTANCES, AND OF
C  STORING THE INFORMATION IN THE FINAL MATRIX THAT WILL SERVE AS THE
C  DATA BASE FROM WHICH THE REPORTS WILL BE ELABORATED.
C
      SUBROUTINE MATCH
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /ROUTE/ RUTENO
      COMMON /BLOCK/ BLOKNO
      COMMON /FECHA/ DATE, MONTH, YEAR
      COMMON /INPUT2/ MACHID, HH, SS, MM, PRSNID
      COMMON /SIZE/ LIMITK, LIMITJ
      COMMON /DISTA1/ FROM1, TO1, DIST1, NAME1
      COMMON /DISTA2/ FROM2, TO2, DIST2, NAME2
      COMMON /DIAS/ M, N
      COMMON /PARAMT/ START, END, INCR
      COMMON /TRIPNO/ AROUND
      COMMON /TRIPS/ POST
      COMMON /PSTREC/ POINTR
      COMMON /STOPNO/ AC
      COMMON /FINAL/ ARRV, DEPT, ON, OFF, LOAD, MEASR, MATCH, IDLE
      COMMON /BLOQUE/ BLOC
      COMMON /TRVL/ TRAVEL
      COMMON /ACUM/ CRUISE, LOADON, IDLETM, VUELTA, SALTO
      COMMON /OK/ OKEY, SKIP
      COMMON /CHKS/ CHK
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
```

```
      INTEGER RUTENO(1201), BLOKNO(1201)
      INTEGER DATE(1201), MONTH(1201), YEAR(1201)
      INTEGER MACHID, HH, SS, MM, PRSNID
      CHARACTER PRNDD
      INTEGER LIMITK, LIMITJ
      INTEGER FROM1(300), TO1(300), DIST1(300)
      INTEGER FROM2(300), TO2(300), DIST2(300)
      CHARACTER*60 NAME1(300), NAME2(300)
      INTEGER M, N
      INTEGER START, END, INCR
      INTEGER AROUND, POST, POINTR(20), AC, B, C
      INTEGER ARRV(300), DEPT(300), ON(300), OFF(300), LOAD(300),
     * MEASR(300), MATCH(300), BLOC(300), IDLE(300)
      INTEGER TRAVEL
      INTEGER CRUISE, LOADON, IDLETM, VUELTA, SALTO
      INTEGER OKEY, SKIP
      INTEGER SIGNAL, TEMPO, ARRIVAL
      INTEGER CHK(8)
C
      POST = 0
      AROUND = 0
      ARRIVAL = 0
C
      CALL LOOPS
C
      SALTO  = 0
      VUELTA = 0
      IDLETM = 0
      LOADON = 0
C
110   CALL LIMITS
C
      DO 105 II = 1, LIMITK - 1
         ARRV(II) = 0
         DEPT(II) = 0
         ON(II)   = 0
         OFF(II)  = 0
         LOAD(II) = 0
         IDLE(II) = 0
         MEASR(II)= 0
         BLOC(II) = 0
         MATCH(II)= 0
105   CONTINUE
      PASS = 1
      CLICK = 11.60
      TRAVEL = 0
C
      DO 200 J = START, END, INCR
        GO TO (200, 220, 203, 230, 221, 210, 220, 220, 200,200,220,200)
     *   TYPE(J)
203     IF (AROUND .GT. 1) THEN
```

```
          IDLETM = IDLETM + 60
        ELSE
          IDLE(AC) = IDLE(AC) + 60
        ENDIF
        GO TO 220
210     IF (AROUND .GT. 1) THEN
          IDLETM = IDLETM + TIME(J) - TIME(J - 1)
        ELSE
          IDLE(AC) = IDLE(AC) + TIME(J) - TIME(J- 1)
        ENDIF
220     TEMP = DIST(J)
221       IF (INCR .EQ. -1) THEN
              TEMP = 1
              DIST(J) = DIST(J) - 1
          ELSE
             TEMP = DIST(J)
          ENDIF
        GO TO (200, 240, 240, 230, 235, 240, 240, 240,200,200,240,200)
    *    TYPE(J)
230     TEMP = 256 + DIST(J)
        GO TO 240
235     IF ( INCR .EQ. -1) THEN
           ARRV(AC) = TIME(J)
        ELSE
           ARRIVAL = TIME(J)
        ENDIF
240     TRAVEL = TRAVEL + TEMP*CLICK
        IF (TYPE(J) .EQ. 2 .OR. TYPE(J) .EQ. 3 .OR. TYPE(J) .EQ. 4
    *    .OR. TYPE(J) .EQ. 6 .OR. TYPE(J) .EQ. 11 .OR. TYPE(J) .EQ. 7)
    *       GO TO 201
 250     IF (INCR .EQ. -1) THEN
             IF (TEMP.NE.1) GO TO 200
         ENDIF
 202    IF(AROUND .EQ. 1 .OR. AROUND .EQ. POST + 1) THEN
             CALL CHEKPT (J)
             IF (OKEY .EQ. 0) GO TO 200
        ENDIF
        TEMPO = DIST1(AC)
        B = TRAVEL - TEMPO
260     IF ( IABS(B) .LE. CLICK) THEN
          IF (AROUND .EQ. 1) AC = AC + INCR
      IF(AROUND .EQ. 1 .OR. AROUND .EQ. POST + 1) THEN
             CALL CHEKPT (J)
             IF (OKEY .EQ. 0) GO TO 200
          ENDIF
          MEASR(AC) = TRAVEL
          ARRV(AC)  = ARRIVAL
          DEPT(AC)  = TIME(J)
          ON(AC)    = ONS(J)
          OFF(AC)   = OFFS(J)
          BLOC(AC)  = BLOKNO(J)
```

```
            IDLE(AC)  = IDLETM
            MATCH(AC) =    1
            TRAVEL    =    0
            IDLETM    =    0
            ARRIVAL   =    0
            IF (AROUND .NE. 1) AC = AC + INCR
          ELSE
            AC = AC + INCR
            IF (AC .GT. LIMITK) THEN
               SALTO = 0
               PASS = PASS + 1
            ENDIF
         IF(AROUND .EQ. 1 .OR. AROUND .EQ. POST + 1) THEN
               CALL CHEKPT (J)
               IF (OKEY .EQ. 0) GO TO 200
            ENDIF
            TEMPO = TEMPO + DIST1(AC)
            C = TRAVEL - TEMPO
            IF ( IABS(B) .GT. IABS(C) ) THEN
               B = C
               GO TO 260
            ELSE
               AC = AC - INCR
               MEASR(AC) = TRAVEL
               ARRV(AC) = ARRIVAL
               DEPT(AC) = TIME(J)
               ON(AC)   = ONS(J)
               OFF(AC)  = OFFS(J)
               BLOC(AC) = BLOKNO(J)
               IDLE(AC) = IDLETM
               MATCH(AC)=     0
               ARRIVAL  =     0
               TRAVEL   =     0
               IDLETM   =     0
               AC = AC + INCR
            ENDIF
          ENDIF
201      IF (INCR .EQ. -1) THEN
            TEMP = DIST(J)
            TRAVEL = TRAVEL + TEMP * CLICK
         ENDIF
200     CONTINUE
C
        IDLETM = IDLE(AC)
        IF (IDLE(LIMITK - 1) .EQ. 0) IDLE(AC) = 0
        CALL PRINT
        IF (AROUND .LT. POST) GO TO 110
300     RETURN
        END
C
C    THIS IS THE SUBROUTINE LOOPS. IT COUNTS HOW MANY TIMES DURING THE
```

```
C   DAY THE BUS WENT AROUND THE SIGNPOST.
C
      SUBROUTINE LOOPS
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /TRIPS/ POST
      COMMON /PSTREC/ POINTR
      COMMON /DIAS/ M, N
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER POST, POINTR(20), M, N
C
      SIGNAL = 1
      IZ = 2
C
      WRITE(*,*) ' M = ', M, ' N = ', N
      DO 100 I = M, N
       IF (TYPE(I) .EQ. 11 .AND. OFFS(I) .EQ. SIGNAL .AND. IZ.EQ. 2) THEN
          POST = POST + 1
          POINTR(POST) = I
          IZ = 0
          GO TO 100
       ENDIF
          IF (TYPE(I) .EQ. 11 .AND. OFFS(I) .EQ. SIGNAL) IZ = IZ + 1
100   CONTINUE
      RETURN
      END
C
C
C     SUBROUTINE LIMITS
C
      SUBROUTINE LIMITS
      COMMON /BLOCK/ BLOKNO
      COMMON /DIAS/ M, N
      COMMON /SIZE/ LIMITK, LIMITJ
      COMMON /PARAMT/ START, END, INCR
      COMMON /TRIPNO/ AROUND
      COMMON /TRIPS/ POST
      COMMON /PSTREC/ POINTR
      COMMON /STOPNO/ AC
      INTEGER BLOKNO(1201)
      INTEGER M, N
      INTEGER LIMITK, LIMITJ
      INTEGER START, END, INCR
      INTEGER AROUND, POST, POINTR(20), AC
C
      IF (AROUND .EQ. 0) GO TO 100
      IF (AROUND .LT. POST) GO TO 200
      GO TO 300
C
100   START = POINTR(1)
      END = M
      INCR = -1
```

```
      AROUND = AROUND + 1
      AC = LIMITK
      GO TO 400
200   IF (BLOKNO(POINTR(AROUND)) .EQ. 18 .AND. AROUND .EQ. POST - 1) GO
     * TO 300
      START = POINTR(AROUND) + 1
      END = POINTR(AROUND + 1)
      INCR = +1
      AROUND = AROUND + 1
      AC = 1
      GO TO 400
300   START = POINTR(AROUND) + 1
      END = N
      INCR = +1
      AC = 1
      AROUND = AROUND + 1
400   RETURN
      END
C
C     SUBROUTINE CHEKPT
C
      SUBROUTINE CHEKPT (J)
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /BLOCK/ BLOKNO
      COMMON /SIZE/ LIMITK, LIMITJ
      COMMON /TRIPNO/ AROUND
      COMMON /TRIPS/ POST
      COMMON /STOPNO/ AC
      COMMON /TRVL/ TRAVEL
      COMMON /ACUM/ CRUISE, LOADON, IDLETM, VUELTA, SALTO
      COMMON /OK/ OKEY, SKIP
      COMMON /CHKS/ CHK
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER BLOKNO(1201)
      INTEGER LIMITK, LIMITJ
      INTEGER AROUND, POST, AC
      INTEGER TRAVEL
      INTEGER CRUISE, LOADON, IDLETM, VUELTA, SALTO
      INTEGER OKEY, SKIP
      INTEGER CHK(8)
C
      OKEY = 1
      CHK(1) = 1
      CHK(2) = 16
      CHK(3) = 45
      CHK(4) = 78
      CHK(5) = 93
      CHK(6) = 111
      CHK(7) = 150
      CHK(8) = 188
C
```

```
      IF (AROUND .EQ. POST + 1) GO TO 6
      IF (BLOKNO(J) .EQ. 18) GO TO 2
      GO TO (1, 2, 3)  (BLOKNO(J) - 14)
1     IF (AC .LE. LIMITK - 1) THEN
          GO TO 4
      ENDIF
      GO TO 5
2     IF (AC .LE. CHK(7) - 1) THEN
          GO TO 4
      ENDIF
      GO TO 5
3     IF (AC .LE. CHK(5) - 1) THEN
          GO TO 4
      ENDIF
      GO TO 5
4     OKEY = 0
      LOADON = LOADON + ONS(J) - OFFS(J)
      CRUISE = TRAVEL
5     GO TO 11
6     IF (BLOKNO(J) .EQ. 18) GO TO 10
      GO TO (7, 8, 9) (BLOKNO(J) -14)
7     IF (AC .GT. CHK(7)) THEN
          GO TO 4
      ENDIF
      GO TO 11
8     IF (AC .GT. CHK(5)) THEN
          GO TO 4
      ENDIF
      GO TO 11
9     IF (AC .GT. CHK(1)) THEN
          GO TO 4
      ENDIF
      GO TO 11
      IF (SKIP .EQ. 1) THEN
           GO TO 11
      ENDIF
10    IF (TIME(J) .GE. 56100) THEN
         AC = CHK(3)
         OKEY = 1
         AROUND = AROUND + 1
         SKIP = 1
         GO TO 11
      ENDIF
      IF (AC .GT. CHK(5)) THEN
         OKEY = 0
         LOADON = LOADON + ONS(J) - OFFS(J)
         CRUISE = TRAVEL
      ENDIF
      SKIP = 0
11    RETURN
```

```
      END
C
C  SUBROUTINE SCHEDL
C
      SUBROUTINE SCHEDL
      COMMON /SKEDUL/ TIMEA
      INTEGER TIMEA(8, 40)
C
      OPEN (UNIT = 8, FILE = ' ')
C
      DO 100 J = 1, 40
          READ(8, 200, END = 120) (TIMEA(K, J), K = 1,8)
200       FORMAT (5X, 8I8)
100   CONTINUE
120   RETURN
      END
C
C     SUBROUTINE PRINT
C
      SUBROUTINE PRINT
      COMMON /DATA/ TYPE, TIME, DIST, ONS, OFFS
      COMMON /ROUTE/ RUTENO
      COMMON /BLOCK/ BLOKNO
      COMMON /FECHA/ DATE, MONTH, YEAR, IDAY
      COMMON /INPUT2/ MACHID, HH, MM, SS, PRSNID
      COMMON /SIZE/ LIMITK, LIMITJ
      COMMON /PARAMT/ START, END, INCR
      COMMON /TRIPNO/ AROUND
      COMMON /FINAL/ ARRV, DEPT, ON, OFF, LOAD, MEASR, MATCH, IDLE
      COMMON /BLOQUE/ BLOC
      COMMON /ACUM/ CRUISE, LOADON, IDLETM, VUELTA, SALTO
      COMMON /HORARO/ SCH, I, CONT
      COMMON /TRVL/ TRAVEL
      COMMON /DISTA1/ FROM1, TO1, DIST1, NAME1
      COMMON /SKEDUL/ TIMEA
      COMMON /CHKS/ CHK
      INTEGER FROM1(300), TO1(300), DIST1(300)
      CHARACTER*60 NAME1(300)
      INTEGER TRAVEL
      INTEGER TYPE(1201), TIME(1201), DIST(1201), ONS(1201), OFFS(1201)
      INTEGER RUTENO(1201), BLOKNO(1201)
      INTEGER DATE(1201), MONTH(1201), YEAR(1201), IDAY(1201)
      INTEGER MACHID, HH, MM, SS
      CHARACTER PRSNID
      INTEGER LIMITK, LIMITJ
      INTEGER START, END, INCR, AROUND
      INTEGER ARRV(300), DEPT(300), ON(300), OFF(300), LOAD(300),
     * MEASR(300), MATCH(300), IDLE(300), BLOC(300)
      CHARACTER*6 CH
      CHARACTER ASTER
      INTEGER CRUISE, LOADON, IDLETM, VUELTA, SALTO
```

```
      INTEGER DISTAN
      INTEGER SCH
      INTEGER CONT
      INTEGER TIMEA(8, 40)
      LOGICAL CAMBIO
      INTEGER CHK(8)
C
      IF (SALTO .EQ. 0 .AND. IKK .EQ. 1) SALTO = 1
      IF (SALTO .EQ. 1) GO TO 105
C
      CALL SCHEDL
      IKK = 1
      JON = START
      WRITE (*, *) 'DO YOU WANT TO PRINT A SAMPLE REPORT ?, YES = 1, NO
     *= 0'
      READ (*, *) IPRINT
      IF (IPRINT .EQ. 0) GO TO 69
      MINUTO = IDLETM/60
      ISEGUN = IDLETM - MINUTO*60
      WRITE (6, 400)
      WRITE (*, 400)
400   FORMAT ('1', 25X, 'ROANOKE VALLEY-METRO APC SYSTEM', ///,
     *  3X, 'ROUTE', 2X, 'BLOCK', 4X, ' NUMBER OF ', 6X,
     *  'DATE', 5X, 'BUS #', 9X, 'ACCUMULATED', /, 19X, 'TIMES AROUND',
     *  /, 54X,'DISTANCE', 2X, 'IDLE', 4X, 'LOAD',1X,'DAY', /)
      WRITE (6, 500) RUTENO(JON), BLOKNO(JON), AROUND, MONTH(JON),
     *  DATE(JON), YEAR(JON), MACHID, REAL(TRAVEL)/5280, MINUTO,
     *  ISEGUN, LOADON, IDAY(JON)
      WRITE (*, 500) RUTENO(JON), BLOKNO(JON), AROUND, MONTH(JON),
     *  DATE(JON), YEAR(JON), MACHID, REAL(TRAVEL)/5280, MINUTO,
     *  ISEGUN, LOADON, IDAY(JON)
500   FORMAT(4X, I3, 4X, I3, 10X, I2, 8X, I2,'/', I2, '/', I2, 4X,
     *  I2, 6X,  F7.2, 3X, I2, ':', I2, 2X, I3, 2X, I3, //)
69    DISTAN = TRAVEL
      SALTO  = 1
      MOMEN = (BLOKNO(START) - 15)*10 + 1
105   CONT   = 1
      IDLETM = 0
C
      IF (IPRINT .EQ. 0) GO TO 68
      WRITE(6, 550)
      WRITE(*, 550)
550   FORMAT ( 1X, 'STOP', 31X, 'ARRIVE', 4X, 'SCHED',
     *  6X, 'ON', 2X, 'OFF', 2X, 'LOAD', 3X,  'DISTANCE', 2X, 'MATCH',
     *  2X, 'IDLE', 4X, 'CHANGE?')
C
68    WRITE (7, 501) RUTENO(JON),BLOKNO(JON),MONTH(JON),DATE(JON),
     *  YEAR(JON),IDAY(JON)
501   FORMAT(1X,I4,2X,2I6,2X,2I4,4X,I3)
C
```

```
      DO 100 I = 1, LIMITK - 1
C
        DISTAN = DISTAN + MEASR(I)
        LOADON = LOADON + ON(I) - OFF(I)
        IDLETM = IDLETM + IDLE(I)
C
        IF (I .EQ. CHK(CONT)) THEN
          SCH = TIMEA(CONT, MOMEN)
C         WRITE (*,*) 'SCH = ',SCH,'   TIMEA =', TIMEA(CONT, MOMEN)
          CONT = CONT + 1
          GO TO 110
        ENDIF
          SCH = 0
110     IF (MATCH(I) .EQ. 1) THEN
          ASTER = '*'
        ELSE
          ASTER = ' '
        ENDIF
        IF (ARRV(I) .EQ. 0 .AND. SCH .EQ. 0) GO TO 99
        IF (ARRV(I) .EQ. 0 .AND. ARRV(I + 1) .EQ. 0) GO TO 99
        IF (BLOC(I) .EQ. BLOC(I+1)) CAMBIO = .FALSE.
        IF (I .EQ. LIMITK - 1) GO TO 120
        IF (CAMBIO) THEN
          CH = 'YES'
          SALTO = 0
        ELSE
          CH = 'NO '
        ENDIF
120   IH1 = ARRV(I)/3600
      IH2 = DEPT(I)/3600
      IH3 = SCH/3600
      IM1 = (ARRV(I) - IH1*3600)/60
      IM2 = (DEPT(I) - IH2*3600)/60
      IM3 = (SCH - IH3*3600)/60
      IS1 = ARRV(I) - IH1*3600 - IM1*60
      IS2 = DEPT(I) - IH2*3600 - IM2*60
      IS3 = SCH - IH3*3600 - IM3*60
        WRITE (6, 575) I, NAME1(I), IH1, IM1, IS1, IH3,
     *   IM3, IS3, ON(I), OFF(I), LOADON, REAL(DISTAN)/5280, ASTER,
     *   IDLE(I), CH
        WRITE (*, 575) I, NAME1(I), IH1, IM1, IS1, IH3,
     *   IM3, IS3, ON(I), OFF(I), LOADON, REAL(DISTAN)/5280, ASTER,
     *  IDLE(I), CH
575     FORMAT (1X, I3, 1X, A28, 2X, I2, ':',
     *   I2, ':', I2, 2X, I2, ':', I2, ':', I2, 2X, 2I4, 4X, I3, 1X,
     *   F10.2, 4X, A1, 3X, I5, 6X, A3)
99      WRITE (7, 585) I,NAME1(I), DEPT(I), SCH, ON(I), OFF(I), LOADON,
     *  DISTAN, RUTENO(JON)
585     FORMAT (1X, I3, 1X, A28, 2X, 2I6, 2X, 2I4, 1X, I3,
```

```
     *  1X, I10, 1X, I2)
C
100    CONTINUE
       IF (IPRINT .EQ. 0) GO TO 67
67     WRITE (7, 605)  LIMITK,RUTENO(JON),IDAY(JON), IDLETM
605    FORMAT (1X, I3, 31X, 2I6, 9X, I5)
       IF (SALTO .EQ. 0) GO TO 610
       SALTO = 1
       MOMEN = MOMEN + 1
610    RETURN
       END
```

APPENDIX B. DATA FILES

Data File 1 (Dec 19, 1985)

```
 1   0   0   0   0
 3   4   34   0   0
 3   4   0   0   0
 6   4   2   0   0
 11   1   11   0   1
 5   3   3   5   2
 3   6   37   0   0
 5   7   109   0   1
 4   11   2   0   0
 5   2   30   1   1
 5   3   94   0   1
 4   5   1   0   0
 5   3   95   1   0
 5   3   116   1   0
 5   4   68   1   0
 5   2   32   2   0
 5   5   134   1   0
 5   5   146   3   0
 4   8   0   0   0
 4   7   0   0   0
 5   2   52   1   0
 5   7   159   0   1
 3   6   62   0   0
 6   14   1   0   0
 5   0   0   3   4
 5   4   38   3   0
 5   5   54   3   0
 3   11   121   0   0
 5   9   137   0   1
 4   6   1   0   0
 4   3   0   0   0
 5   8   142   0   1
 5   6   231   1   0
 4   5   0   0   0
 5   4   78   0   1
 4   6   1   0   0
 4   4   0   0   0
 11   1   54   0   1
 4   5   1   0   0
 3   8   179   0   0
 5   1   1   1   0
 5   2   49   0   1
 5   3   33   3   0
 5   4   94   1   0
 4   6   2   0   0
 4   7   3   0   0
 5   3   113   0   1
```

```
 5   2   30   1   0
 4   5   0   0   0
 5   4   64   0   1
 4   5   0   0   0
 3   7   50   0   0
 5   0   2   2   1
 4   8   2   0   0
 4   6   0   0   0
 4   7   0   0   0
 5   8   106   1   0
 3   11   44   0   0
 5   4   2   5   1
 5   5   122   0   1
 4   5   0   0   0
 4   5   1   0   0
 5   14   241   0   1
 5   3   38   1   0
 5   2   78   0   1
 4   8   2   0   0
 4   3   3   0   0
 4   8   2   0   0
 4   8   0   0   0
 5   5   55   2   0
 3   8   106   0   0
 6   10   2   0   0
 11   5   0   0   1
 5   0   1   6   5
 3   5   46   0   0
 5   8   90   1   0
 4   5   0   0   0
 5   4   62   1   0
 5   2   56   1   0
 5   6   148   5   0
 5   4   112   1   0
 5   2   34   1   0
 5   3   86   1   0
 5   5   184   1   0
 4   6   0   0   0
 5   3   58   0   1
 5   6   102   2   0
 4   6   2   0   0
 4   9   2   0   0
 3   12   165   0   0
 5   1   1   4   4
 5   5   38   2   1
 5   7   56   6   0
 5   10   214   1   0
 4   10   0   0   0
 4   4   1   0   0
 5   9   145   0   1
 5   2   36   0   1
```

```
 4  7  1  0  0
 5  9  215  1  0
 4  6  1  0  0
 4  6  0  0  0
 5  1  7  1  1
 4  5  1  0  0
 11  8  247  0  1
 3  2  0  0  0
 5  1  249  0  1
 5  5  154  3  0
 4  8  1  0  0
 4  4  0  0  0
 5  3  145  1  0
 4  6  2  0  0
 5  3  44  1  0
 5  5  116  1  0
 4  4  5  0  0
 5  3  48  1  0
 4  11  5  0  0
 5  5  195  1  0
 5  5  167  0  1
 5  3  44  1  0
 3  9  156  0  0
 4  8  1  0  0
 4  6  0  0  0
 5  4  171  1  0
 4  10  3  0  0
 4  6  0  0  0
 4  6  3  0  0
 4  6  0  0  0
 4  6  0  0  0
 4  8  1  0  0
 5  4  67  0  1
 3  5  42  0  0
 5  3  2  1  3
11 5  5  0  1
 7  5  0  41  108
 8  0  0  72  121
12 12 19 85 1
12 4 U 16 7
```

Data File 2 (Dec 24, 1985)

```
 1   0   0   0   0
 3   4   0   0   0
 3   4   0   0   0
 6   3   4   0   0
 4   4   0   0   0
 11   2   4   0   1
 3   4   0   0   0
 5   3   92  1   0
 5   4   105   1   0
 4   6   0   0   0
 4   6   1   0   0
 4   6   1   0   0
 5   4   118   1   0
 4   5   4   0   0
 5   5   152   1   0
 5   2   40   1   0
 4   7   0   0   0
 4   8   0   0   0
 5   5   104   0   1
 3   8   65   0   0
 3   5   0   0   0
 6   10   0   0   0
 5   6   1   0   1
 5   4   38   1   0
 5   5   54   4   0
 4   10   0   0   0
 4   8   2   0   0
 5   5   1   1   0
 4   0   2   0   0
 5   4   55   2   0
 4   8   1   0   0
 4   5   2   0   0
 5   5   133   1   0
 4   5   0   0   0
 4   4   2   0   0
 4   7   0   0   0
 11   4   165   0   1
 3   4   0   0   0
 5   1   1   1   1
 5   6   156   0   5
 5   8   226   1   0
 4   5   2   0   0
 5   6   102   1   0
 5   3   70   2   0
 5   3   36   2   0
 5   5   56   2   0
 5   3   90   2   0
 5   2   38   2   0
 4   6   0   0   0
```

```
 5   7   248   2   0
 3   4   19   0   0
 4   7   2   0   0
 4   5   1   0   0
 5   3   66   1   0
 5   7   228   0   1
 5   6   57   1   4
 3   5   40   0   0
 6   10   1   0   0
 5   0   0   3   0
 5   9   146   1   0
 4   5   2   0   0
 4   5   0   0   0
 3   5   26   0   0
 4   10   0   0   0
 4   6   0   0   0
 4   5   2   0   0
 5   5   206   1   0
 4   8   1   0   0
 5   5   123   0   2
 5   6   40   1   6
 3   8   106   0   0
 11   1   13   0   1
 6   21   2   0   0
 5   0   0   1   5
 4   14   0   0   0
 4   5   0   0   0
 4   6   0   0   0
 4   5   0   0   0
 4   5   1   0   0
 5   4   102   2   0
 5   3   106   1   0
 4   6   0   0   0
 4   7   1   0   0
 3   12   162   0   0
 6   24   1   0   0
 5   0   0   2   4
 5   5   38   3   0
 5   7   52   3   0
 4   9   0   0   0
 5   2   38   1   0
 3   7   160   0   0
 4   5   1   0   0
 5   7   163   0   1
 4   5   1   0   0
 5   0   11   0   1
 4   7   0   0   0
 4   5   0   0   0
 4   4   1   0   0
 4   4   0   0   0
 11   6   242   0   1
```

```
3  3  0  0  0
6  15  1  0  0
5  0  0  1  1
5  7  156  5  2
5  6  180  2  0
4  4  1  0  0
5  5  221  0  1
5  6  224  0  1
4  7  0  0  0
4  5  1  0  0
3  4  9  0  0
5  3  116  1  0
4  5  0  0  0
5  4  156  0  2
5  3  72  0  1
3  11  196  0  0
5  1  2  0  1
3  13  96  0  0
5  4  2  11  0
3  4  2  0  0
5  3  2  1  0
5  7  146  3  0
4  6  0  0  0
4  5  1  0  0
5  6  133  1  0
5  2  46  0  1
5  3  78  1  0
4  6  1  0  0
4  6  0  0  0
4  6  0  0  0
5  9  111  0  1
5  9  240  2  1
5  5  42  0  6
5  3  36  0  1
11  4  66  0  1
3  3  0  0  0
6  10  2  0  0
5  0  0  6  1
4  14  0  0  0
4  5  1  0  0
5  4  153  0  1
5  3  114  1  0
5  3  74  1  0
5  5  198  1  0
4  5  1  0  0
5  3  91  1  0
3  4  2  0  0
5  10  102  2  0
4  5  0  0  0
5  11  172  0  1
3  14  250  0  0
```

```
 6  5  2  0  0
 5  0  0  3  5
 3  13  90  0  0
 5  1  2  7  0
 4  12  0  0  0
 4  7  1  0  0
 5  5  255  1  0
 5  2  58  0  1
 5  5  120  1  0
 5  2  24  0  1
 4  5  0  0  0
 5  4  108  0  2
 5  7  190  0  2
 4  6  0  0  0
 5  3  122  0  1
 4  5  2  0  0
 11  3  134  0  1
 5  7  110  1  1
 5  3  122  0  1
 4  7  0  0  0
 5  1  6  1  0
 5  4  156  1  0
 5  5  202  1  0
 5  2  68  1  0
 5  7  224  1  0
 5  3  116  0  1
 4  5  0  0  0
 5  5  130  3  0
 4  7  1  0  0
 4  7  0  0  0
 5  1  35  0  1
 4  9  0  0  0
 5  2  22  1  0
 5  4  52  0  1
 3  6  39  0  0
 5  1  1  6  0
 3  4  0  0  0
 6  12  2  0  0
 5  0  0  4  2
 5  2  14  2  0
 5  5  82  0  1
 4  8  1  0  0
 4  7  0  0  0
 5  3  71  0  1
 5  6  156  1  1
 4  6  1  0  0
 5  6  93  0  1
 4  5  1  0  0
 5  8  213  0  2
 4  6  1  0  0
 5  7  121  0  2
```

```
 3   7   38   0   0
 5   1   2   6   4
 11   8   104   0   1
 3   1   0   0   0
 5   2   2   3   0
 5   14   244   0   1
 5   10   210   1   0
 5   3   90   0   1
 4   5   1   0   0
 5   2   13   1   0
 4   6   0   0   0
 4   6   1   0   0
 5   2   43   0   1
 5   4   30   2   0
 5   3   62   1   0
 4   7   1   0   0
 4   8   0   0   0
 5   2   65   0   3
 5   14   140   2   3
 5   4   36   3   0
 5   3   4   0   1
 5   3   46   3   0
 5   12   220   1   0
 4   8   0   0   0
 5   4   228   1   0
 4   12   0   0   0
 4   8   0   0   0
 4   6   0   0   0
 5   1   10   0   2
 5   3   52   1   1
 4   7   0   0   0
 5   1   18   0   1
 5   6   238   0   1
 4   5   0   0   0
 3   5   37   0   0
 11   2   0   0   1
 6   11   1   0   0
 5   7   150   2   0
 5   4   84   3   0
 5   5   90   0   1
 4   11   1   0   0
 4   5   0   0   0
 5   8   239   1   0
 4   7   0   0   0
 4   5   0   0   0
 5   2   26   1   0
 5   10   138   0   2
 5   7   252   1   0
 4   7   1   0   0
 5   8   175   0   2
 3   11   101   0   0
```

```
6  7  1  0  0
5  0  0  8  0
5  5  50  0  1
4  9  0  0  0
4  6  1  0  0
4  8  0  0  0
4  5  0  0  0
4  8  0  0  0
4  4  0  0  0
5  7  145  0  1
4  8  0  0  0
3  7  51  0  0
5  1  1  2  1
3  7  41  0  0
6  19  1  0  0
5  0  0  5  8
3  9  107  0  0
11  8  0  0  1
6  3  1  0  0
5  0  0  13  2
4  16  0  0  0
5  9  167  2  0
5  2  24  1  0
5  7  212  1  0
4  6  1  0  0
5  2  19  1  0
4  5  0  0  0
4  7  1  0  0
5  3  51  2  0
4  6  0  0  0
5  8  204  1  0
3  6  108  0  0
5  1  2  0  1
3  6  112  0  0
5  0  2  2  7
5  5  34  3  0
5  8  54  1  1
4  12  0  0  0
4  6  1  0  0
4  3  0  0  0
5  6  63  1  0
5  5  82  0  1
5  2  62  0  1
5  4  94  0  1
4  8  0  0  0
5  2  62  0  1
5  5  88  0  1
5  4  52  1  2
4  5  1  0  0
5  2  17  0  1
4  7  0  0  0
```

```
 4  5  0  0  0
 11  3  95  0  1
 5  9  157  2  1
 4  8  0  0  0
 4  5  0  0  0
 5  3  144  0  1
 5  7  224  1  0
 4  8  1  0  0
 5  1  15  0  1
 4  6  0  0  0
 4  5  1  0  0
 5  3  109  1  0
 5  6  156  2  0
 4  9  0  0  0
 5  6  82  0  2
 3  5  37  0  0
 6  9  1  0  0
 5  0  0  3  1
 4  11  0  0  0
 4  5  1  0  0
 4  7  0  0  0
 4  6  1  0  0
 4  7  0  0  0
 4  4  1  0  0
 5  4  85  0  1
 4  5  1  0  0
 3  12  203  0  0
 11  14  0  0  1
 6  3  2  0  0
 5  0  0  3  4
 7  2  0  107  37
 8  0  0  181  253
12 12 24 85 1
12 2 U 17 100
```

Data File 3 (Jan 3, 1986)

```
 1   0   0   0   0
 3   5   0   0   0
 3   4   0   0   0
 3   6   0   0   0
 6   2   0   0   0
 4   8   0   0   0
 11   0   1   0   1
 6   17   1   0   0
 5   0   0   19   1
 5   2   18   1   0
 5   3   82   0   1
 5   6   131   1   0
 4   8   0   0   0
 4   6   0   0   0
 5   4   99   0   1
 5   2   42   0   1
 4   5   0   0   0
 5   1   20   0   1
 4   6   0   0   0
 4   7   0   0   0
 5   2   62   0   1
 4   6   0   0   0
 5   7   96   0   4
 11   6   105   0   1
 3   1   0   0   0
 6   30   1   0   0
 5   0   0   3   8
 4   8   1   0   0
 4   4   2   0   0
 4   4   2   0   0
 4   5   0   0   0
 4   6   7   0   0
 5   4   110   1   0
 4   8   0   0   0
 4   6   1   0   0
 4   11   0   0   0
 3   4   10   0   0
 6   6   1   0   0
 5   0   0   3   2
 5   10   94   4   0
 4   10   0   0   0
 5   3   38   0   1
 4   6   1   0   0
 4   7   0   0   0
 4   6   0   0   0
 5   8   235   0   1
 4   6   1   0   0
 4   5   6   0   0
 4   6   1   0   0
```

```
 4   5   2   0   0
 11   2   72   0   1
 3   4   0   0   0
 6   13   2   0   0
 5   5   72   4   0
 5   4   82   1   0
 4   7   0   0   0
 4   5   3   0   0
 4   5   0   0   0
 5   7   113   2   0
 4   6   1   0   0
 5   6   205   1   0
 5   4   56   1   1
 5   3   28   2   0
 4   6   2   0   0
 5   7   244   1   0
 5   3   26   1   0
 5   12   250   0   1
 11   8   89   0   1
 3   3   0   0   0
 6   5   1   0   0
 5   0   0   13   2
 5   3   22   1   0
 5   2   36   0   1
 5   3   44   0   1
 5   5   90   0   1
 4   8   1   0   0
 5   6   237   0   1
 4   6   2   0   0
 5   6   248   1   0
 5   6   162   0   1
 4   6   1   0   0
 4   5   0   0   0
 5   8   173   0   1
 5   5   42   0   4
 3   9   106   0   0
 11   3   0   0   1
 6   21   2   0   0
 5   0   0   5   6
 5   10   245   0   1
 4   5   1   0   0
 5   5   162   1   0
 4   5   2   0   0
 4   5   1   0   0
 4   6   0   0   0
 4   6   0   0   0
 4   11   0   0   0
 3   13   231   0   0
 6   9   2   0   0
 5   0   0   3   3
 5   5   36   5   0
```

```
 5   8   54   4   0
 3   11   116   0   0
 4   9   2   0   0
 5   2   40   0   1
 4   4   0   0   0
 5   4   100   0   1
 5   6   56   0   1
 4   7   2   0   0
 5   1   18   1   0
 5   2   38   0   1
 5   3   40   0   1
 5   2   38   0   1
 5   7   252   0   1
 5   3   158   0   1
 5   4   168   0   1
 4   8   1   0   0
 4   5   5   0   0
 11   4   87   0   1
 3   3   0   0   0
 6   13   1   0   0
 5   3   72   1   0
 5   5   86   2   2
 4   6   3   0   0
 4   5   1   0   0
 4   5   2   0   0
 4   7   1   0   0
 5   3   43   1   0
 5   3   76   1   0
 5   5   156   1   0
 4   9   1   0   0
 4   5   0   0   0
 4   7   1   0   0
 5   12   184   2   1
 11   5   41   0   1
 3   3   0   0   0
 6   5   1   0   0
 5   0   0   5   0
 5   5   16   2   0
 4   9   0   0   0
 4   6   1   0   0
 5   5   153   1   0
 4   10   0   0   0
 4   6   0   0   0
 4   5   0   0   0
 4   8   0   0   0
 4   10   2   0   0
 5   6   74   3   2
 3   7   41   0   0
 5   1   1   3   2
 11   8   102   0   1
 3   1   0   0   0
```

```
 5   4   2   1   1
 5   8   130   0   1
 5   2   32   0   1
 4   10   2   0   0
 5   3   96   1   0
 5   5   148   1   0
 4   5   0   0   0
 5   6   162   1   0
 5   3   34   0   1
 5   3   92   1   0
 5   5   156   0   1
 5   4   136   1   0
 4   6   0   0   0
 5   6   130   0   1
 5   3   40   1   0
 3   13   252   0   0
 6   9   2   0   0
 5   0   0   2   3
 5   6   36   3   1
 5   5   56   5   0
 4   13   0   0   0
 5   2   40   1   0
 5   3   80   0   1
 5   4   78   1   0
 4   4   1   0   0
 4   11   0   0   0
 5   0   3   0   1
 5   11   247   0   2
 4   5   2   0   0
 4   6   2   0   0
 4   5   0   0   0
 4   5   1   0   0
11   4   77   0   1
 3   2   0   0   0
 5   1   1   0   1
 5   5   76   2   0
 5   5   72   5   1
 5   2   48   2   0
 5   3   34   1   0
 4   7   0   0   0
 4   5   0   0   0
 4   7   0   0   0
 5   7   228   1   0
 5   2   38   1   0
 5   2   38   0   1
 4   4   0   0   0
 5   3   50   0   1
 4   6   3   0   0
 5   8   155   1   0
 4   9   1   0   0
 3   14   193   0   0
```

```
 5  1  11  0  1
 11  4  42  0  1
 7  2  0  65  142
 8  0  0  109 82
12 1 3 86 1
12 5 U 14 46
```

APPENDIX C. BUS STOP DISTANCE DATA

210	1	1140	BRENDLES (Time Point)	
1	2	1414	OGDEN & FRANKLIN ROAD	
2	3	386	FRANKLIN RD. & ELM VIEW RD.	
3	4	428	FRANKLIN RD. & AVENHAM MANOR	
4	5	496	FRANKLIN RD. & TANGLEWOOD	(Wendy's)
5	6	2170	FRANKLIN RD. & PENARTH 1	
6	7	293	FRANKLIN RD. & PENARTH 2	
7	8	369	FRANKLIN RD. & TOWNSIDE	
8	9	386	FRANKLIN RD. & K-MART	
9	10	342	FRANKLIN RD. & DUKE OF GLOUC	
10	11	379	FRANKLIN RD. & WILLOW OAK	
11	12	454	AVENHAM & DILLARD	
12	13	533	AVENHAM & WHITE OAK (2)	
13	14	659	AVENHAM &	(Midblock)
14	15	665	AVENHAM & CLYDESDALE	
15	16	406	AVENHAM & CASSELL	
16	17	468	AVENHAM & AUDUBON	
17	18	543	AVENHAM & 29 th	
18	19	460	AVENHAM & 28 th	
19	20	457	AVENHAM & 27 th	
20	21	454	AVENHAM & 26 th	
21	22	372	26 th & WYCLIFFE	
22	23	380	26 th & CAROLINA	
23	24	396	26 th & ROSALIND	
24	25	397	26 th & CRYSTAL SPRING	
25	26	302	26 th & RICHELIEU	
26	27	390	26 th & JEFFERSON	
27	28	471	JEFFERSON & 25 th	
28	29	441	JEFFERSON & 24 th	
29	30	471	JEFFERSON & 23 rd	
30	31	458	JEFFERSON & 22 nd	
31	32	376	JEFFERSON & YELLOW MT.	
32	33	748	JEFFERSON & WELLER	
33	34	460	ROANOKE MEMORIAL HOSPITAL	
34	35	738	BELLEVIEW & ASH	(RMH Parking)
35	36	528	BELLEVIEW & THYME	
36	37	313	BELLEVIEW & LINDEN	
37	38	740	BELLEVIEW & LAUREL	
38	39	694	BELLEVIEW & WALNUT	
39	40	465	WALNUT AVENUE BRIDGE 2	
40	41	1047	WALNUT AVENUE BRIDGE 1	
41	42	420	JEFFERSON & MAPLE	
42	43	419	JEFFERSON & ALBEMARLE	
43	44	463	JEFFERSON & HIGHLAND	
44	45	459	JEFFERSON & MOUNTAIN	
45	46	344	JEFFERSON & ELM	
46	47	570	JEFFERSON & BULLIT	(Public Library)
47	48	344	JEFFERSON & FRANKLIN RD.	

48	49	435	JEFFERSON & CHURCH	
49	50	431	JEFFERSON & CAMPBELL	
50	51	597	CAMPBELL & MARKET ST.	(Fox Bargain Store)
51	52	1084	WILLIAMSON ROAD & SHENANDOAH	
52	53	689	WELLS & COMMONWEALTH	
53	54	439	WELLS & HOTEL ROANOKE 1	
54	55	349	WELLS & JEFFERSON	
55	56	347	JEFFERSON & GILMER	
56	57	161	GILMER & GAINSBORO	
57	58	402	GAINSBORO & PATTON	
58	59	217	GAINSBORO & HARRISON	
59	60	729	GAINSBORO & MADISON	
60	61	493	GAINSBORO & McDOWELL	
61	62	725	GAINSBORO & ORANGE	(On Orange)
62	63	588	ORANGE & QUARRY	
63	64	901	ORANGE & 5 th	
64	65	699	ORANGE & LUCY ADDISON JR. H. S	
65	66	375	ORANGE & 8 th	
66	67	883	ORANGE & 10 th	
67	68	658	ORANGE & 11 th	(On 11 th)
68	69	202	11 th & HANNOVER	
69	70	339	11 th & STAUNTON	
70	71	415	STAUNTON & 12 th	
71	72	452	STAUNTON & 13 th	
72	73	289	STAUNTON & 14 th	
73	74	333	STAUNTON & EUREKA CIRCLE 1	
74	75	284	STAUNTON & 15 th	
75	76	457	STAUNTON & 16 th	
76	77	457	STAUNTON & 17 th	
77	78	456	STAUNTON & 18 th	
78	79	460	STAUNTON & 19 th	
79	80	456	STAUNTON & 20 th	
80	81	451	STAUNTON & 21 st	
81	82	454	STAUNTON & 22 nd	
82	83	436	STAUNTON & 23 rd	
83	84	552	STAUNTON & 24 th	(Maryland)
84	85	601	STAUNTON & LAFAYETTE	
85	86	456	LAFAYETTE & NEW YORK	(B&G Grocery)
86	87	226	LAFAYETTE & CLIFTON	
87	88	382	LAFAYETTE & DELAWARE	
88	89	393	LAFAYETTE & MASSACHUSSETS	
89	90	844	LAFAYETTE & FLORIDA	
90	91	434	LAFAYETTE &	(Midblock)
91	92	442	LAFAYETTE & COVE ROAD	
92	93	607	COVE ROAD & FAIRLAND	
93	94	514	COVE ROAD & ASPEN	
94	95	336	COVE ROAD & WELLSLEY	
95	96	339	COVE ROAD & ABBOTT	
96	97	505	COVE ROAD & COVELAND	
97	98	696	COVE ROAD & GOLFSIDE	
98	99	314	COVE ROAD & GUILDHALL	

99	100	569	COVE ROAD & HERSHBERGER	
100	101	1403	COVE ROAD &	(Midblock, # 3437)
101	102	764	COVE ROAD &	(Midblock, # 3540)
102	103	871	COVE ROAD & ROUTT	(On Routt)
103	104	390	ROUTT & BLACK OAK	
104	105	273	ROUTT & FERNCLIFF	
105	106	615	FERNCLIFF APARTMENTS N.	(Wm. Fleming H.S.)
106	107	842	FERNCLIFF APARTMENTS S.	(Wm. Fleming H.S.)
107	108	566	FERNCLIFF &	(Wm. Ruffner Jr. H.S.)
108	109	408	FERNCLIFF &	(Hills Dept. Store)
109	110	782	FERNCLIFF & HERSHBERGER	
110	111	295	HERSHBERGER & SWARTHMORE	
111	112	1026	HERSHBERGER & COVE ROAD	
112	113	567	COVE ROAD & GUILDHALL	
113	114	316	COVE ROAD & GOLFSIDE	
114	115	695	COVE ROAD & FRESNO	
115	116	522	COVE ROAD & ABBOTT	
116	117	297	COVE ROAD & WELLSLEY	
117	118	359	COVE ROAD & ASPEN	
118	119	504	COVE ROAD & FAIRLAND	
119	120	688	COVE ROAD & LAFAYETTE	
120	121	384	LAFAYETTE &	(Midblock, # 1529)
121	122	409	LAFAYETTE & FLORIDA	
122	123	805	LAFAYETTE & MASSACHUSSETS	
123	124	447	LAFAYETTE & DELAWARE	
124	125	338	LAFAYETTE & CLIFTON	
125	126	350	LAFAYETTE & NEW YORK	
126	127	441	LAFAYETTE & STAUNTON	(On Staunton)
127	128	617	STAUNTON & 24 th	(Maryland)
128	129	449	STAUNTON & 23 rd	
129	130	450	STAUNTON & 22 nd	
130	131	455	STAUNTON & 21 st	
131	132	455	STAUNTON & 20 th	
132	133	458	STAUNTON & 19 th	
133	134	448	STAUNTON & 18 th	
134	135	461	STAUNTON & 17 th	
135	136	450	STAUNTON & 16 th	
136	137	445	STAUNTON & 15 th	
137	138	355	STAUNTON & EUREKA CIRCLE 1	
138	139	357	STAUNTON & 14 th	
139	140	214	STAUNTON & 13 th	
140	141	455	STAUNTON & 12 th	
141	142	407	STAUNTON & 11 th	
142	143	438	11 th & HANOVER	(Farside stop)
143	144	575	ORANGE & 10 th	
144	145	299	ORANGE &	(Midblock)
145	146	746	ORANGE & 8 th	
146	147	592	ORANGE & LUCY ADDISON JR.H.S	
147	148	678	ORANGE & 5 th	
148	149	736	ORANGE & QUARRY	
149	150	947	ORANGE & GAINSBORO	(On Gainsboro)

150	151	278	GAINSBORO & McDOWELL	
151	152	484	GAINSBORO & MADISON	
152	153	759	GAINSBORO & HARRISON	
153	154	304	GAINSBORO & PATTON	
154	155	309	1 st & GILMER	
155	156	313	GILMER & JEFFERSON	
156	157	254	JEFFERSON & WELLS	
157	158	587	WELLS & HOTEL ROANOKE 1	
158	159	186	WELLS & HOTEL ROANOKE 2	
159	160	311	WELLS & WILLIAMSON ROAD	
160	161	393	WILLIAMSON ROAD & SHENANDOAH	
161	162	567	WILLIAMSON ROAD & SALEM	
162	163	305	SALEM & 1 st	
163	164	650	SALEM & JEFFERSON	
164	165	394	JEFFERSON & CAMPBELL	
165	166	265	JEFFERSON & CHURCH	(Opposite Heironimus)
166	167	439	JEFFERSON & FRANKLIN RD.	(U.S. Recruitment Cntr.)
167	168	275	JEFFERSON & BULLIT	
168	169	550	JEFFERSON & ELM	
169	170	282	JEFFERSON & MOUNTAIN	
170	171	457	JEFFERSON & HIGHLAND	
171	172	464	JEFFERSON & ALBEMARLE	
172	173	470	JEFFERSON & MAPLE	
173	174	536	WALNUT AVENUE BRIDGE 1	
174	175	971	WALNUT AVENUE BRIDGE 2	
175	176	482	WALNUT & BELLEVIEW	
176	177	616	BELLEVIEW & LAUREL	
177	178	441	BELLEVIEW & PINK	
178	179	307	BELLEVIEW & LINDEN	
179	180	878	BELLEVIEW & ASH	(RMH Parking Lot)
180	181	583	ROANOKE MEMORIAL HOSPITAL	
181	182	899	JEFFERSON & WELLER	
182	183	399	JEFFERSON & MC CLANAHAN	
183	184	538	JEFFERSON & 22 nd	
184	185	467	JEFFERSON & 23 rd	
185	186	466	JEFFERSON & 24 th	
186	187	457	JEFFERSON & 25 th	
187	188	444	JEFFERSON & 26 th	
188	189	376	26 th & RICHELIEU	
189	190	303	26 th & CRYSTAL SPRING	
190	191	395	26 th & ROSALIND	
191	192	391	26 th & CAROLINA	
192	193	404	26 th & WYCLIFF	
193	194	384	26 th & AVENHAM	
194	195	476	AVENHAM & 27 th	
195	196	461	AVENHAM & 28 th	
196	197	499	AVENHAM & 29 th	
197	198	469	AVENHAM & WILDWOOD	
198	199	522	AVENHAM & CASSELL	
199	200	373	AVENHAM & WHITE OAK (1)	
200	201	692	AVENHAM &	(Midblock)

201	202	613	AVENHAM & WHITE OAK (2)	
202	203	928	AVENHAM & FRANKLIN ROAD	
203	204	469	FRANKLIN RD. & DUKE OF GLOUC	
204	205	350	FRANKLIN RD. & K-MART	
205	206	463	FRANKLIN RD. & TOWNSIDE	
206	207	812	FRANKLIN RD. & PENARTH	(DMV Sign)
207	208	3184	SUPER X	
208	209	619	PENNEY'S	
209	210	50	SIGNBOARD RECORD	

APPENDIX D. SCHEDULE DATA

Schedule Time For Dec. 19 Data

	20400	20820	21600	22320	22800	23280	24000	24720
B	25500	25920	26700	27600	28200	28680	29400	30120
L	30900	31320	32100	33000	33600	34080	34800	35520
O	36300	36720	37500	38400	39000	39480	40200	40920
C	41700	42120	42900	43800	44400	44880	45900	46620
K	47400	47820	48600	49500	50100	50580	51300	52020
	52800	53220	54000	54900	55500	55980	56700	57420
1	58200	58620	59400	60300	60900	61380	62400	63120
5	63900	64320	65100	65880	66300	66780	67500	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	21000	21720
B	22200	22620	23400	24120	24600	25080	25800	26520
L	27300	27720	28500	29400	30000	30480	31200	31920
O	32700	33120	33900	34800	35400	35880	36600	37320
C	38100	38520	39300	40200	40800	41280	42000	43020
K	43800	44220	45000	45900	46500	46980	47700	48420
	49200	49620	50400	51300	51900	52380	53100	54120
1	54900	55320	56400	57300	57900	58380	59100	59820
6	60600	61020	61800	6270U	63300	63780	64500	65220
	66000	66420	67200	67800	68100	0	0	0
	0	0	0	0	0	0	0	0
B	48600	49500	50100	50580	51300	52020	52800	53220
L	54000	54900	55500	55980	56700	57420	58200	58620
O	59400	60300	60900	61380	62400	63120	63900	64320
C	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	23100	23820
B	24600	25020	0	25020	25500	25980	26700	27420
L	28200	28620	0	28620	29100	29580	30300	31020
O	31800	32220	0	32220	32700	0	0	0
C	0	0	0	56100	56700	57180	57900	58620
K	59400	59820	60600	61500	62100	62700	56400	0
	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Schedule Time For Dec. 24 Data

	20400	20820	21600	22320	22800	23280	24000	24720
B	25500	25920	26700	27600	28200	28680	29400	30120
L	30900	31320	32100	33000	33600	34080	34800	35520
O	36300	36720	37500	38400	39000	39480	40200	40920
C	41700	42120	42900	43800	44400	44880	45900	46620
K	47400	47820	48600	49500	50100	50580	51300	52020
	52800	53220	54000	54900	55500	55980	56700	57420
1	58200	58620	59400	60300	60900	61380	62400	63120
5	63900	64320	65100	65880	66300	66780	67500	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	21000	21720
B	22200	22620	23400	24120	24600	25080	25800	26520
L	27300	27720	28500	29400	30000	30480	31200	31920
O	32700	33120	33900	34800	35400	35880	36600	37320
C	38100	38520	39300	40200	40800	41280	42000	43020
K	43800	44220	45000	45900	46500	46980	47700	48420
	49200	49620	50400	51300	51900	52380	53100	54120
1	54900	55320	56400	57300	57900	58380	59100	59820
6	60600	61020	61800	62700	63300	63780	64500	65220
	66000	66420	67200	67800	68100	0	0	0
	0	0	0	0	0	0	0	0
B	33900	34800	35400	35880	36600	37320	38100	38520
L	39300	40200	40800	41280	42000	43020	43800	44220
O	45000	45900	46500	46980	47700	48420	49200	49620
C	50400	51300	51900	52380	53100	54120	54900	55320
K	56400	57300	57900	58380	59100	59820	60600	61020
	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	23100	23820
B	24600	25020	0	25020	25500	25980	26700	27420
L	28200	28620	0	28620	29100	29580	30300	31020
O	31800	32220	0	32220	32700	0	0	0
C	0	0	0	56100	56700	57180	57900	58620
K	59400	59820	60600	61500	62100	62700	56400	0
	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Schedule Time For Jan. 3 Data

	20400	20820	21600	22320	22800	23280	24000	24720
B	25500	25920	26700	27600	28200	28680	29400	30120
L	30900	31320	32100	33000	33600	34080	34800	35520
O	36300	36720	37500	38400	39000	39480	40200	40920
C	41700	42120	42900	43800	44400	44880	45900	46620
K	47400	47820	48600	49500	50100	50580	51300	52020
	52800	53220	54000	54900	55500	55980	56700	57420
1	58200	58620	59400	60300	60900	61380	62400	63120
5	63900	64320	65100	65880	66300	66780	67500	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	21000	21720
B	22200	22620	23400	24120	24600	25080	25800	26520
L	27300	27720	28500	29400	30000	30480	31200	31920
O	32700	33120	33900	34800	35400	35880	36600	37320
C	38100	38520	39300	40200	40800	41280	42000	43020
K	43800	44220	45000	45900	46500	46980	47700	48420
	49200	49620	50400	51300	51900	52380	53100	54120
1	54900	55320	56400	57300	57900	58380	59100	59820
6	60600	61020	61800	62700	63300	63780	64500	65220
	66000	66420	67200	67800	68100	0	0	0
	0	0	0	0	0	0	0	0
B	36300	36720	37500	38400	39000	39480	40200	40920
L	41700	42120	42900	43800	44400	44880	45900	46620
O	47400	47820	48600	49500	50100	50580	51300	52020
C	52800	53220	54000	54900	55500	55980	56700	57420
K	58200	58620	59400	60300	60900	61380	62400	63120
	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	23100	23820
B	24600	25020	0	25020	25500	25980	26700	27420
L	28200	28620	0	28620	29100	29580	30300	31020
O	31800	32220	0	32220	32700	0	0	0
C	0	0	0	56100	56700	57180	57900	58620
K	59400	59820	60600	61500	62100	62700	56400	0
	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

APPENDIX E. VEHICLE PROFILE FILE (RAW DATA)

1	17	12	19	85	1					
1	BRENDLES (Time Point)				48690	48600	5	2	3	437	1
2	OGDEN & FRANKLIN ROAD				0	0	0	0	3	437	1
3	FRANKLIN RD. & ELM VIEW R				48885	0	0	1	2	2130	1
4	FRANKLIN RD. & AVENHAM MA				0	0	0	0	2	2130	1
5	FRANKLIN RD. & TANGLEWOOD				0	0	0	0	2	2130	1
6	FRANKLIN RD. & PENARTH 1				0	0	0	0	2	2130	1
7	FRANKLIN RD. & PENARTH 2				49080	0	1	1	2	5470	1
8	FRANKLIN RD. & TOWNSIDE				0	0	0	0	2	5470	1
9	FRANKLIN RD. & K-MART				0	0	0	0	2	5470	1
10	FRANKLIN RD. & DUKE OF GL				49125	0	0	1	1	6560	1
11	FRANKLIN RD. & WILLOW OAK				0	0	0	0	1	6560	1
12	AVENHAM & DILLARD				0	0	0	0	1	6560	1
13	AVENHAM & WHITE OAK (2)				0	0	0	0	1	6560	1
14	AVENHAM &				0	0	0	0	1	6560	1
15	AVENHAM & CLYDESDALE				0	0	0	0	1	6560	1
16	AVENHAM & CASSELL				0	0	0	0	1	6560	1
17	AVENHAM & AUDUBON				0	0	0	0	1	6560	1
18	AVENHAM & 29 th				49245	0	1	0	2	10643	1
19	AVENHAM & 28 th				0	0	0	0	2	10643	1
20	AVENHAM & 27 th				0	0	0	0	2	10643	1
21	AVENHAM & 26 th				49290	0	1	0	3	11988	1
22	26 th & WYCLIFFE				0	0	0	0	3	11988	1
23	26 th & CAROLINA				49350	0	1	0	4	12776	1
24	26 th & ROSALIND				49380	0	2	0	6	13147	1
25	26 th & CRYSTAL SPRING				0	0	0	0	6	13147	1
26	26 th & RICHELIEU				0	0	0	0	6	13147	1
27	26 th & JEFFERSON				0	0	0	0	6	13147	1
28	JEFFERSON & 25 th				49455	0	1	0	7	14701	1
29	JEFFERSON & 24 th				0	0	0	0	7	14701	1
30	JEFFERSON & 23 rd				0	0	0	0	7	14701	1
31	JEFFERSON & 22 nd				0	0	0	0	7	14701	1
32	JEFFERSON & YELLOW MT.				49530	0	3	0	10	16394	1
33	JEFFERSON & WELLER				0	0	0	0	10	16394	1
34	ROANOKE MEMORIAL HOSPITAL				0	49500	0	0	10	16394	1
35	BELLEVIEW & ASH				0	0	0	0	10	16394	1
36	BELLEVIEW & THYME				0	0	0	0	10	16394	1
37	BELLEVIEW & LINDEN				0	0	0	0	10	16394	1
38	BELLEVIEW & LAUREL				0	0	0	0	10	16394	1
39	BELLEVIEW & WALNUT				0	0	0	0	10	16394	1
40	WALNUT AVENUE BRIDGE 2				0	0	0	0	10	16394	1
41	WALNUT AVENUE BRIDGE 1				0	0	0	0	10	16394	1
42	JEFFERSON & MAPLE				0	0	0	0	10	16394	1
43	JEFFERSON & ALBEMARLE				49785	0	1	0	11	22935	1
44	JEFFERSON & HIGHLAND				0	0	0	0	11	22935	1
45	JEFFERSON & MOUNTAIN				0	0	0	0	11	22935	1
46	JEFFERSON & ELM				0	0	0	0	11	22935	1
47	JEFFERSON & BULLIT				49890	0	0	1	10	24779	1

48	JEFFERSON & FRANKLIN RD.	0	0	0	0	10	24779	1
49	JEFFERSON & CHURCH	50190	50100	3	4	9	25509	1
50	JEFFERSON & CAMPBELL	50250	0	3	0	12	25949	1
51	CAMPBELL & MARKET ST.	50325	0	3	0	15	26575	1
52	WILLIAMSON ROAD & SHENAND	0	0	0	0	15	26575	1
53	WELLS & COMMONWEALTH	0	0	0	0	15	26575	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	15	26575	1
55	WELLS & JEFFERSON	0	0	0	0	15	26575	1
56	JEFFERSON & GILMER	0	0	0	0	15	26575	1
57	GILMER & GAINSBORO	50625	0	0	1	14	29567	1
58	GAINSBORO & PATTON	0	0	0	0	14	29567	1
59	GAINSBORO & HARRISON	0	0	0	0	14	29567	1
60	GAINSBORO & MADISON	0	0	0	0	14	29567	1
61	GAINSBORO & McDOWELL	0	0	0	0	14	29567	1
62	GAINSBORO & ORANGE	0	0	0	0	14	29567	1
63	ORANGE & QUARRY	0	0	0	0	14	29567	1
64	ORANGE & 5 th	0	0	0	0	14	29567	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	14	29567	1
66	ORANGE & 8 th	0	0	0	0	14	29567	1
67	ORANGE & 10 th	0	50580	0	0	14	29567	1
68	ORANGE & 11 th	0	0	0	0	14	29567	1
69	11 th & HANNOVER	0	0	0	0	14	29567	1
70	11 th & STAUNTON	0	0	0	0	14	29567	1
71	STAUNTON & 12 th	50880	0	0	1	13	37164	1
72	STAUNTON & 13 th	0	0	0	0	13	37164	1
73	STAUNTON & 14 th	0	0	0	0	13	37164	1
74	STAUNTON & EUREKA CIRCLE	0	0	0	0	13	37164	1
75	STAUNTON & 15 th	0	0	0	0	13	37164	1
76	STAUNTON & 16 th	0	0	0	0	13	37164	1
77	STAUNTON & 17 th	0	0	0	0	13	37164	1
78	STAUNTON & 18 th	50970	0	1	0	14	39843	1
79	STAUNTON & 19 th	0	0	0	0	14	39843	1
80	STAUNTON & 20 th	0	0	0	0	14	39843	1
81	STAUNTON & 21 st	0	0	0	0	14	39843	1
82	STAUNTON & 22 nd	0	0	0	0	14	39843	1
83	STAUNTON & 23 rd	0	0	0	0	14	39843	1
84	STAUNTON & 24 th	0	0	0	0	14	39843	1
85	STAUNTON & LAFAYETTE	0	0	0	0	14	39843	1
86	LAFAYETTE & NEW YORK	51105	0	0	1	13	43716	1
87	LAFAYETTE & CLIFTON	0	0	0	0	13	43716	1
88	LAFAYETTE & DELAWARE	0	0	0	0	13	43716	1
89	LAFAYETTE & MASSACHUSSETS	0	0	0	0	13	43716	1
90	LAFAYETTE & FLORIDA	0	0	0	0	13	43716	1
91	LAFAYETTE &	0	0	0	0	13	43716	1
92	LAFAYETTE & COVE ROAD	0	0	0	0	13	43716	1
93	COVE ROAD & FAIRLAND	0	0	0	0	13	43716	1
94	COVE ROAD & ASPEN	0	0	0	0	13	43716	1
95	COVE ROAD & WELLSLEY	0	0	0	0	13	43716	1
96	COVE ROAD & ABBOTT	0	0	0	0	13	43716	1
97	COVE ROAD & COVELAND	0	0	0	0	13	43716	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	13	43716	1

99	COVE ROAD & GUILDHALL	0	0	0	0	13	43716	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	13	43716	1
101	COVE ROAD &	0	0	0	0	13	43716	1
102	COVE ROAD &	0	0	0	0	13	43716	1
103	COVE ROAD & ROUTT	0	0	0	0	13	43716	1
104	ROUTT & BLACK OAK	0	0	0	0	13	43716	1
105	ROUTT & FERNCLIFF	0	0	0	0	13	43716	1
106	FERNCLIFF APARTMENTS N.	0	51300	0	0	13	43716	1
107	FERNCLIFF APARTMENTS S.	51480	0	1	0	14	55360	1
108	FERNCLIFF &	51510	0	0	1	13	55928	1
109	FERNCLIFF &	51555	0	3	0	16	56310	1
110	FERNCLIFF & HERSHBERGER	0	0	0	0	16	56310	1
111	HERSHBERGER & SWARTHMORE	51615	0	1	0	17	57400	1
112	HERSHBERGER & COVE ROAD	0	0	0	0	17	57400	1
113	COVE ROAD & GUILDHALL	0	0	0	0	17	57400	1
114	COVE ROAD & GOLFSIDE	0	0	0	0	17	57400	1
115	COVE ROAD & FRESNO	0	0	0	0	17	57400	1
116	COVE ROAD & ABBOTT	0	0	0	0	17	57400	1
117	COVE ROAD & WELLSLEY	0	0	0	0	17	57400	1
118	COVE ROAD & ASPEN	0	0	0	0	17	57400	1
119	COVE ROAD & FAIRLAND	0	0	0	0	17	57400	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	17	57400	1
121	LAFAYETTE &	0	0	0	0	17	57400	1
122	LAFAYETTE & FLORIDA	0	0	0	0	17	57400	1
123	LAFAYETTE & MASSACHUSSETS	0	0	0	0	17	57400	1
124	LAFAYETTE & DELAWARE	0	0	0	0	17	57400	1
125	LAFAYETTE & CLIFTON	51855	0	0	1	16	64706	1
126	LAFAYETTE & NEW YORK	51885	0	1	0	17	65054	1
127	LAFAYETTE & STAUNTON	0	0	0	0	17	65054	1
128	STAUNTON & 24 th	0	0	0	0	17	65054	1
129	STAUNTON & 23 rd	0	0	0	0	17	65054	1
130	STAUNTON & 22 nd	0	0	0	0	17	65054	1
131	STAUNTON & 21 st	0	0	0	0	17	65054	1
132	STAUNTON & 20 th	0	0	0	0	17	65054	1
133	STAUNTON & 19 th	0	0	0	0	17	65054	1
134	STAUNTON & 18 th	52020	0	0	1	16	68765	1
135	STAUNTON & 17 th	0	0	0	0	16	68765	1
136	STAUNTON & 16 th	0	0	0	0	16	68765	1
137	STAUNTON & 15 th	0	0	0	0	16	68765	1
138	STAUNTON & EUREKA CIRCLE	0	0	0	0	16	68765	1
139	STAUNTON & 14 th	0	0	0	0	16	68765	1
140	STAUNTON & 13 th	0	0	0	0	16	68765	1
141	STAUNTON & 12 th	0	0	0	0	16	68765	1
142	STAUNTON & 11 th	0	0	0	0	16	68765	1
143	11 th & HANOVER	52200	0	2	1	17	72337	1
144	ORANGE & 10 th	0	52020	0	0	17	72337	1
145	ORANGE &	0	0	0	0	17	72337	1
146	ORANGE & 8 th	0	0	0	0	17	72337	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	17	72337	1
148	ORANGE & 5 th	0	0	0	0	17	72337	1
149	ORANGE & QUARRY	0	0	0	0	17	72337	1

150	ORANGE & GAINSBORO	0	0	0	0	17	72337	1
151	GAINSBORO & McDOWELL	0	0	0	0	17	72337	1
152	GAINSBORO & MADISON	0	0	0	0	17	72337	1
153	GAINSBORO & HARRISON	0	0	0	0	17	72337	1
154	GAINSBORO & PATTON	0	0	0	0	17	72337	1
155	1 st & GILMER	0	0	0	0	17	72337	1
156	GILMER & JEFFERSON	0	0	0	0	17	72337	1
157	JEFFERSON & WELLS	0	0	0	0	17	72337	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	17	72337	1
159	WELLS & HOTEL ROANOKE 2	0	0	0	0	17	72337	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	17	72337	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	17	72337	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	17	72337	1
163	SALEM & 1 st	0	0	0	0	17	72337	1
164	SALEM & JEFFERSON	52635	0	1	0	18	82496	1
165	JEFFERSON & CAMPBELL	0	0	0	0	18	82496	1
166	JEFFERSON & CHURCH	52860	52800	5	1	22	83029	1
167	JEFFERSON & FRANKLIN RD.	0	0	0	0	22	83029	1
168	JEFFERSON & BULLIT	0	0	0	0	22	83029	1
169	JEFFERSON & ELM	0	0	0	0	22	83029	1
170	JEFFERSON & MOUNTAIN	52935	0	0	1	21	84444	1
171	JEFFERSON & HIGHLAND	0	0	0	0	21	84444	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	21	84444	1
173	JEFFERSON & MAPLE	0	0	0	0	21	84444	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	21	84444	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	21	84444	1
176	WALNUT & BELLEVIEW	0	0	0	0	21	84444	1
177	BELLEVIEW & LAUREL	0	0	0	0	21	84444	1
178	BELLEVIEW & PINK	0	0	0	0	21	84444	1
179	BELLEVIEW & LINDEN	0	0	0	0	21	84444	1
180	BELLEVIEW & ASH	0	0	0	0	21	84444	1
181	ROANOKE MEMORIAL HOSPITAL	0	53220	0	0	21	84444	1
182	JEFFERSON & WELLER	0	0	0	0	21	84444	1
183	JEFFERSON & MC CLANAHAN	0	0	0	0	21	84444	1
184	JEFFERSON & 22 nd	0	0	0	0	21	84444	1
185	JEFFERSON & 23 rd	0	0	0	0	21	84444	1
186	JEFFERSON & 24 th	53295	0	0	1	20	93189	1
187	JEFFERSON & 25 th	53340	0	1	0	21	93629	1
188	JEFFERSON & 26 th	0	0	0	0	21	93629	1
189	26 th & RICHELIEU	53370	0	0	1	20	94533	1
190	26 th & CRYSTAL SPRING	0	0	0	0	20	94533	1
191	26 th & ROSALIND	0	0	0	0	20	94533	1
192	26 th & CAROLINA	0	0	0	0	20	94533	1
193	26 th & WYCLIFF	0	0	0	0	20	94533	1
194	26 th & AVENHAM	0	0	0	0	20	94533	1
195	AVENHAM & 27 th	0	0	0	0	20	94533	1
196	AVENHAM & 28 th	0	0	0	0	20	94533	1
197	AVENHAM & 29 th	0	0	0	0	20	94533	1
198	AVENHAM & WILDWOOD	0	0	0	0	20	94533	1
199	AVENHAM & CASSELL	0	0	0	0	20	94533	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	20	94533	1

201	AVENHAM &	0	0	0	0	20	94533	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	20	94533	1
203	AVENHAM & FRANKLIN ROAD	0	0	0	0	20	94533	1
204	FRANKLIN RD. & DUKE OF GL	0	0	0	0	20	94533	1
205	FRANKLIN RD. & K-MART	0	0	0	0	20	94533	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	20	94533	1
207	FRANKLIN RD. & PENARTH	0	0	0	0	20	94533	1
208	SUPER X	0	0	0	0	20	94533	1
209	PENNEY'S	53850	0	2	0	22	107128	1
210		1	1			750		
1	17 12 19 85	1						
1	BRENDLES (Time Point)	54195	54000	6	5	23	107139	1
2	OGDEN & FRANKLIN ROAD	54390	0	1	0	24	108716	1
3	FRANKLIN RD. & ELM VIEW R	0	0	0	0	24	108716	1
4	FRANKLIN RD. & AVENHAM MA	0	0	0	0	24	108716	1
5	FRANKLIN RD. & TANGLEWOOD	0	0	0	0	24	108716	1
6	FRANKLIN RD. & PENARTH 1	0	0	0	0	24	108716	1
7	FRANKLIN RD. & PENARTH 2	54525	0	1	0	25	112404	1
8	FRANKLIN RD. & TOWNSIDE	0	0	0	0	25	112404	1
9	FRANKLIN RD. & K-MART	54555	0	1	0	26	113053	1
10	FRANKLIN RD. & DUKE OF GL	0	0	0	0	26	113053	1
11	FRANKLIN RD. & WILLOW OAK	0	0	0	0	26	113053	1
12	AVENHAM & DILLARD	0	0	0	0	26	113053	1
13	AVENHAM & WHITE OAK (2)	54645	0	5	0	31	114769	1
14	AVENHAM &	0	0	0	0	31	114769	1
15	AVENHAM & CLYDESDALE	54705	0	1	0	32	116068	1
16	AVENHAM & CASSELL	54735	0	1	0	33	116462	1
17	AVENHAM & AUDUBON	0	0	0	0	33	116462	1
18	AVENHAM & 29 th	54780	0	1	0	34	117459	1
19	AVENHAM & 28 th	0	0	0	0	34	117459	1
20	AVENHAM & 27 th	0	0	0	0	34	117459	1
21	AVENHAM & 26 th	0	0	0	0	34	117459	1
22	26 th & WYCLIFFE	0	0	0	0	34	117459	1
23	26 th & CAROLINA	54855	0	1	0	35	119593	1
24	26 th & ROSALIND	0	0	0	0	35	119593	1
25	26 th & CRYSTAL SPRING	0	0	0	0	35	119593	1
26	26 th & RICHELIEU	0	0	0	0	35	119593	1
27	26 th & JEFFERSON	0	0	0	0	35	119593	1
28	JEFFERSON & 25 th	0	0	0	0	35	119593	1
29	JEFFERSON & 24 th	0	0	0	0	35	119593	1
30	JEFFERSON & 23 rd	0	0	0	0	35	119593	1
31	JEFFERSON & 22 nd	0	0	0	0	35	119593	1
32	JEFFERSON & YELLOW MT.	54990	0	0	1	34	123234	1
33	JEFFERSON & WELLER	0	0	0	0	34	123234	1
34	ROANOKE MEMORIAL HOSPITAL	55080	54900	2	0	36	124417	1
35	BELLEVIEW & ASH	0	0	0	0	36	124417	1
36	BELLEVIEW & THYME	0	0	0	0	36	124417	1
37	BELLEVIEW & LINDEN	0	0	0	0	36	124417	1
38	BELLEVIEW & LAUREL	0	0	0	0	36	124417	1
39	BELLEVIEW & WALNUT	0	0	0	0	36	124417	1
40	WALNUT AVENUE BRIDGE 2	0	0	0	0	36	124417	1

41	WALNUT AVENUE BRIDGE 1	0	0	0	0	36	124417	1
42	JEFFERSON & MAPLE	0	0	0	0	36	124417	1
43	JEFFERSON & ALBEMARLE	0	0	0	0	36	124417	1
44	JEFFERSON & HIGHLAND	0	0	0	0	36	124417	1
45	JEFFERSON & MOUNTAIN	0	0	0	0	36	124417	1
46	JEFFERSON & ELM	0	0	0	0	36	124417	1
47	JEFFERSON & BULLIT	0	0	0	0	36	124417	1
48	JEFFERSON & FRANKLIN RD.	0	0	0	0	36	124417	1
49	JEFFERSON & CHURCH	55500	55500	4	4	36	132326	1
50	JEFFERSON & CAMPBELL	55575	0	2	1	37	132766	1
51	CAMPBELL & MARKET ST.	55680	0	6	0	43	133415	1
52	WILLIAMSON ROAD & SHENAND	0	0	0	0	43	133415	1
53	WELLS & COMMONWEALTH	0	0	0	0	43	133415	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	43	133415	1
55	WELLS & JEFFERSON	55830	0	1	0	44	135897	1
56	JEFFERSON & GILMER	0	0	0	0	44	135897	1
57	GILMER & GAINSBORO	0	0	0	0	44	135897	1
58	GAINSBORO & PATTON	0	0	0	0	44	135897	1
59	GAINSBORO & HARRISON	0	0	0	0	44	135897	1
60	GAINSBORO & MADISON	0	0	0	0	44	135897	1
61	GAINSBORO & McDOWELL	0	0	0	0	44	135897	1
62	GAINSBORO & ORANGE	0	0	0	0	44	135897	1
63	ORANGE & QUARRY	0	0	0	0	44	135897	1
64	ORANGE & 5 th	0	0	0	0	44	135897	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	44	135897	1
66	ORANGE & 8 th	0	0	0	0	44	135897	1
67	ORANGE & 10 th	0	55980	0	0	44	135897	1
68	ORANGE & 11 th	0	0	0	0	44	135897	1
69	11 th & HANNOVER	0	0	0	0	44	135897	1
70	11 th & STAUNTON	56175	0	0	1	43	143529	1
71	STAUNTON & 12 th	56205	0	0	1	42	143946	1
72	STAUNTON & 13 th	0	0	0	0	42	143946	1
73	STAUNTON & 14 th	0	0	0	0	42	143946	1
74	STAUNTON & EUREKA CIRCLE	0	0	0	0	42	143946	1
75	STAUNTON & 15 th	0	0	0	0	42	143946	1
76	STAUNTON & 16 th	0	0	0	0	42	143946	1
77	STAUNTON & 17 th	0	0	0	0	42	143946	1
78	STAUNTON & 18 th	0	0	0	0	42	143946	1
79	STAUNTON & 19 th	0	0	0	0	42	143946	1
80	STAUNTON & 20 th	0	0	0	0	42	143946	1
81	STAUNTON & 21 st	0	0	0	0	42	143946	1
82	STAUNTON & 22 nd	0	0	0	0	42	143946	1
83	STAUNTON & 23 rd	0	0	0	0	42	143946	1
84	STAUNTON & 24 th	56445	0	1	0	43	149421	1
85	STAUNTON & LAFAYETTE	0	0	0	0	43	149421	1
86	LAFAYETTE & NEW YORK	0	0	0	0	43	149421	1
87	LAFAYETTE & CLIFTON	0	0	0	0	43	149421	1
88	LAFAYETTE & DELAWARE	0	0	0	0	43	149421	1
89	LAFAYETTE & MASSACHUSSETS	0	0	0	0	43	149421	1
90	LAFAYETTE & FLORIDA	0	0	0	0	43	149421	1
91	LAFAYETTE &	0	0	0	0	43	149421	1

92	LAFAYETTE & COVE ROAD	0	0	0	0	43	149421	1
93	COVE ROAD & FAIRLAND	0	0	0	0	43	149421	1
94	COVE ROAD & ASPEN	0	0	0	0	43	149421	1
95	COVE ROAD & WELLSLEY	0	0	0	0	43	149421	1
96	COVE ROAD & ABBOTT	0	0	0	0	43	149421	1
97	COVE ROAD & COVELAND	56640	0	1	1	43	155452	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	43	155452	1
99	COVE ROAD & GUILDHALL	0	0	0	0	43	155452	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	43	155452	1
101	COVE ROAD &	0	0	0	0	43	155452	1
102	COVE ROAD &	0	0	0	0	43	155452	1
103	COVE ROAD & ROUTT	0	0	0	0	43	155452	1
104	ROUTT & BLACK OAK	0	0	0	0	43	155452	1
105	ROUTT & FERNCLIFF	0	0	0	0	43	155452	1
106	FERNCLIFF APARTMENTS N.	0	56700	0	0	43	155452	1
107	FERNCLIFF APARTMENTS S.	0	0	0	0	43	155452	1
108	FERNCLIFF &	0	0	0	0	43	155452	1
109	FERNCLIFF &	0	0	0	0	43	155452	1
110	FERNCLIFF & HERSHBERGER	0	0	0	0	43	155452	1
111	HERSHBERGER & SWARTHMORE	56880	0	0	1	42	164186	1
112	HERSHBERGER & COVE ROAD	0	0	0	0	42	164186	1
113	COVE ROAD & GUILDHALL	0	0	0	0	42	164186	1
114	COVE ROAD & GOLFSIDE	56955	0	3	0	45	165972	1
115	COVE ROAD & FRESNO	0	0	0	0	45	165972	1
116	COVE ROAD & ABBOTT	0	0	0	0	45	165972	1
117	COVE ROAD & WELLSLEY	0	0	0	0	45	165972	1
118	COVE ROAD & ASPEN	0	0	0	0	45	165972	1
119	COVE ROAD & FAIRLAND	0	0	0	0	45	165972	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	45	165972	1
121	LAFAYETTE &	0	0	0	0	45	165972	1
122	LAFAYETTE & FLORIDA	0	0	0	0	45	165972	1
123	LAFAYETTE & MASSACHUSSETS	0	0	0	0	45	165972	1
124	LAFAYETTE & DELAWARE	0	0	0	0	45	165972	1
125	LAFAYETTE & CLIFTON	0	0	0	0	45	165972	1
126	LAFAYETTE & NEW YORK	0	0	0	0	45	165972	1
127	LAFAYETTE & STAUNTON	0	0	0	0	45	165972	1
128	STAUNTON & 24 th	0	0	0	0	45	165972	1
129	STAUNTON & 23 rd	0	0	0	0	45	165972	1
130	STAUNTON & 22 nd	57180	0	1	0	46	173604	1
131	STAUNTON & 21 st	0	0	0	0	46	173604	1
132	STAUNTON & 20 th	0	0	0	0	46	173604	1
133	STAUNTON & 19 th	0	0	0	0	46	173604	1
134	STAUNTON & 18 th	0	0	0	0	46	173604	1
135	STAUNTON & 17 th	0	0	0	0	46	173604	1
136	STAUNTON & 16 th	0	0	0	0	46	173604	1
137	STAUNTON & 15 th	0	0	0	0	46	173604	1
138	STAUNTON & EUREKA CIRCLE	57315	0	1	0	47	177106	1
139	STAUNTON & 14 th	0	0	0	0	47	177106	1
140	STAUNTON & 13 th	0	0	0	0	47	177106	1
141	STAUNTON & 12 th	0	0	0	0	47	177106	1
142	STAUNTON & 11 th	57390	0	1	0	48	178451	1

143	11 th & HANOVER	0	0	0	0	48	178451	1
144	ORANGE & 10 th	0	57420	0	0	48	178451	1
145	ORANGE &	0	0	0	0	48	178451	1
146	ORANGE & 8 th	0	0	0	0	48	178451	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	48	178451	1
148	ORANGE & 5 th	57495	0	1	0	49	182034	1
149	ORANGE & QUARRY	0	0	0	0	49	182034	1
150	ORANGE & GAINSBORO	0	0	0	0	49	182034	1
151	GAINSBORO & McDOWELL	0	0	0	0	49	182034	1
152	GAINSBORO & MADISON	0	0	0	0	49	182034	1
153	GAINSBORO & HARRISON	0	0	0	0	49	182034	1
154	GAINSBORO & PATTON	0	0	0	0	49	182034	1
155	1 st & GILMER	0	0	0	0	49	182034	1
156	GILMER & JEFFERSON	0	0	0	0	49	182034	1
157	JEFFERSON & WELLS	0	0	0	0	49	182034	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	49	182034	1
159	WELLS & HOTEL ROANOKE 2	57735	0	1	0	50	187323	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	50	187323	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	50	187323	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	50	187323	1
163	SALEM & 1 st	0	0	0	0	50	187323	1
164	SALEM & JEFFERSON	57810	0	0	1	49	189260	1
165	JEFFERSON & CAMPBELL	57855	0	1	0	50	189770	1
166	JEFFERSON & CHURCH	0	58200	0	0	50	189770	1
167	JEFFERSON & FRANKLIN RD.	0	0	0	0	50	189770	1
168	JEFFERSON & BULLIT	0	0	0	0	50	189770	1
169	JEFFERSON & ELM	0	0	0	0	50	189770	1
170	JEFFERSON & MOUNTAIN	0	0	0	0	50	189770	1
171	JEFFERSON & HIGHLAND	0	0	0	0	50	189770	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	50	189770	1
173	JEFFERSON & MAPLE	0	0	0	0	50	189770	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	50	189770	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	50	189770	1
176	WALNUT & BELLEVIEW	0	0	0	0	50	189770	1
177	BELLEVIEW & LAUREL	0	0	0	0	50	189770	1
178	BELLEVIEW & PINK	0	0	0	0	50	189770	1
179	BELLEVIEW & LINDEN	0	0	0	0	50	189770	1
180	BELLEVIEW & ASH	0	0	0	0	50	189770	1
181	ROANOKE MEMORIAL HOSPITAL	0	58620	0	0	50	189770	1
182	JEFFERSON & WELLER	0	0	0	0	50	189770	1
183	JEFFERSON & MC CLANAHAN	0	0	0	0	50	189770	1
184	JEFFERSON & 22 nd	58260	0	1	0	51	199512	1
185	JEFFERSON & 23 rd	0	0	0	0	51	199512	1
186	JEFFERSON & 24 th	0	0	0	0	51	199512	1
187	JEFFERSON & 25 th	0	0	0	0	51	199512	1
188	JEFFERSON & 26 th	0	0	0	0	51	199512	1
189	26 th & RICHELIEU	0	0	0	0	51	199512	1
190	26 th & CRYSTAL SPRING	0	0	0	0	51	199512	1
191	26 th & ROSALIND	0	0	0	0	51	199512	1
192	26 th & CAROLINA	0	0	0	0	51	199512	1
193	26 th & WYCLIFF	0	0	0	0	51	199512	1

194	26 th & AVENHAM	0	0	0	0	51	199512	1
195	AVENHAM & 27 th	0	0	0	0	51	199512	1
196	AVENHAM & 28 th	0	0	0	0	51	199512	1
197	AVENHAM & 29 th	0	0	0	0	51	199512	1
198	AVENHAM & WILDWOOD	0	0	0	0	51	199512	1
199	AVENHAM & CASSELL	0	0	0	0	51	199512	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	51	199512	1
201	AVENHAM &	0	0	0	0	51	199512	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	51	199512	1
203	AVENHAM & FRANKLIN ROAD	0	0	0	0	51	199512	1
204	FRANKLIN RD. & DUKE OF GL	0	0	0	0	51	199512	1
205	FRANKLIN RD. & K-MART	0	0	0	0	51	199512	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	51	199512	1
207	FRANKLIN RD. & PENARTH	0	0	0	0	51	199512	1
208	SUPER X	0	0	0	0	51	199512	1
209	PENNEY'S	0	0	0	0	51	199512	1
210		1	1			990		

1	17	12	24	85	2						
1	BRENDLES (Time Point)				34065	33900	1	0	1	4124	1
2	OGDEN & FRANKLIN ROAD				34125	0	1	0	2	5342	1
3	FRANKLIN RD. & ELM VIEW R				0	0	0	0	2	5342	1
4	FRANKLIN RD. & AVENHAM MA				0	0	0	0	2	5342	1
5	FRANKLIN RD. & TANGLEWOOD				0	0	0	0	2	5342	1
6	FRANKLIN RD. & PENARTH 1				0	0	0	0	2	5342	1
7	FRANKLIN RD. & PENARTH 2				0	0	0	0	2	5342	1
8	FRANKLIN RD. & TOWNSIDE				0	0	0	0	2	5342	1
9	FRANKLIN RD. & K-MART				0	0	0	0	2	5342	1
10	FRANKLIN RD. & DUKE OF GL				0	0	0	0	2	5342	1
11	FRANKLIN RD. & WILLOW OAK				0	0	0	0	2	5342	1
12	AVENHAM & DILLARD				0	0	0	0	2	5342	1
13	AVENHAM & WHITE OAK (2)				0	0	0	0	2	5342	1
14	AVENHAM &				0	0	0	0	2	5342	1
15	AVENHAM & CLYDESDALE				0	0	0	0	2	5342	1
16	AVENHAM & CASSELL				0	0	0	0	2	5342	1
17	AVENHAM & AUDUBON				0	0	0	0	2	5342	1
18	AVENHAM & 29 th				0	0	0	0	2	5342	1
19	AVENHAM & 28 th				0	0	0	0	2	5342	1
20	AVENHAM & 27 th				0	0	0	0	2	5342	1
21	AVENHAM & 26 th				34455	0	1	0	3	15641	1
22	26 th & WYCLIFFE				0	0	0	0	3	15641	1
23	26 th & CAROLINA				0	0	0	0	3	15641	1
24	26 th & ROSALIND				0	0	0	0	3	15641	1
25	26 th & CRYSTAL SPRING				0	0	0	0	3	15641	1
26	26 th & RICHELIEU				0	0	0	0	3	15641	1
27	26 th & JEFFERSON				0	0	0	0	3	15641	1
28	JEFFERSON & 25 th				0	0	0	0	3	15641	1
29	JEFFERSON & 24 th				0	0	0	0	3	15641	1
30	JEFFERSON & 23 rd				0	0	0	0	3	15641	1
31	JEFFERSON & 22 nd				0	0	0	0	3	15641	1
32	JEFFERSON & YELLOW MT.				34605	0	1	0	4	20420	1
33	JEFFERSON & WELLER				34635	0	1	0	5	20884	1
34	ROANOKE MEMORIAL HOSPITAL				0	34800	0	0	5	20884	1
35	BELLEVIEW & ASH				0	0	0	0	5	20884	1
36	BELLEVIEW & THYME				0	0	0	0	5	20884	1
37	BELLEVIEW & LINDEN				0	0	0	0	5	20884	1
38	BELLEVIEW & LAUREL				0	0	0	0	5	20884	1
39	BELLEVIEW & WALNUT				0	0	0	0	5	20884	1
40	WALNUT AVENUE BRIDGE 2				0	0	0	0	5	20884	1
41	WALNUT AVENUE BRIDGE 1				0	0	0	0	5	20884	1
42	JEFFERSON & MAPLE				0	0	0	0	5	20884	1
43	JEFFERSON & ALBEMARLE				0	0	0	0	5	20884	1
44	JEFFERSON & HIGHLAND				0	0	0	0	5	20884	1
45	JEFFERSON & MOUNTAIN				0	0	0	0	5	20884	1
46	JEFFERSON & ELM				34935	0	0	1	4	28028	1
47	JEFFERSON & BULLIT				0	0	0	0	4	28028	1
48	JEFFERSON & FRANKLIN RD.				35370	0	0	1	3	28793	1
49	JEFFERSON & CHURCH				35430	35400	1	0	4	29233	1
50	JEFFERSON & CAMPBELL				35505	0	4	0	8	29859	1

51	CAMPBELL & MARKET ST.	0	0	0	0	8	29859	1
52	WILLIAMSON ROAD & SHENAND	0	0	0	0	8	29859	1
53	WELLS & COMMONWEALTH	0	0	0	0	8	29859	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	8	29859	1
55	WELLS & JEFFERSON	0	0	0	0	8	29859	1
56	JEFFERSON & GILMER	0	0	0	0	8	29859	1
57	GILMER & GAINSBORO	0	0	0	0	8	29859	1
58	GAINSBORO & PATTON	0	0	0	0	8	29859	1
59	GAINSBORO & HARRISON	0	0	0	0	8	29859	1
60	GAINSBORO & MADISON	0	0	0	0	8	29859	1
61	GAINSBORO & McDOWELL	0	0	0	0	8	29859	1
62	GAINSBORO & ORANGE	35850	0	1	0	9	35831	1
63	ORANGE & QUARRY	0	0	0	0	9	35831	1
64	ORANGE & 5 th	0	0	0	0	9	35831	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	9	35831	1
66	ORANGE & 8 th	0	0	0	0	9	35831	1
67	ORANGE & 10 th	35910	35880	2	0	11	39461	1
68	ORANGE & 11 th	0	0	0	0	11	39461	1
69	11 th & HANNOVER	0	0	0	0	11	39461	1
70	11 th & STAUNTON	0	0	0	0	11	39461	1
71	STAUNTON & 12 th	0	0	0	0	11	39461	1
72	STAUNTON & 13 th	0	0	0	0	11	39461	1
73	STAUNTON & 14 th	0	0	0	0	11	39461	1
74	STAUNTON & EUREKA CIRCLE	0	0	0	0	11	39461	1
75	STAUNTON & 15 th	0	0	0	0	11	39461	1
76	STAUNTON & 16 th	0	0	0	0	11	39461	1
77	STAUNTON & 17 th	0	0	0	0	11	39461	1
78	STAUNTON & 18 th	0	0	0	0	11	39461	1
79	STAUNTON & 19 th	0	0	0	0	11	39461	1
80	STAUNTON & 20 th	0	0	0	0	11	39461	1
81	STAUNTON & 21 st	0	0	0	0	11	39461	1
82	STAUNTON & 22 nd	0	0	0	0	11	39461	1
83	STAUNTON & 23 rd	0	0	0	0	11	39461	1
84	STAUNTON & 24 th	0	0	0	0	11	39461	1
85	STAUNTON & LAFAYETTE	36180	0	1	0	12	46976	1
86	LAFAYETTE & NEW YORK	0	0	0	0	12	46976	1
87	LAFAYETTE & CLIFTON	0	0	0	0	12	46976	1
88	LAFAYETTE & DELAWARE	0	0	0	0	12	46976	1
89	LAFAYETTE & MASSACHUSSETS	0	0	0	0	12	46976	1
90	LAFAYETTE & FLORIDA	0	0	0	0	12	46976	1
91	LAFAYETTE &	0	0	0	0	12	46976	1
92	LAFAYETTE & COVE ROAD	0	0	0	0	12	46976	1
93	COVE ROAD & FAIRLAND	0	0	0	0	12	46976	1
94	COVE ROAD & ASPEN	0	0	0	0	12	46976	1
95	COVE ROAD & WELLSLEY	0	0	0	0	12	46976	1
96	COVE ROAD & ABBOTT	0	0	0	0	12	46976	1
97	COVE ROAD & COVELAND	0	0	0	0	12	46976	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	12	46976	1
99	COVE ROAD & GUILDHALL	0	0	0	0	12	46976	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	12	46976	1
101	COVE ROAD &	0	0	0	0	12	46976	1

102	COVE ROAD &	0	0	0	0	12	46976	1
103	COVE ROAD & ROUTT	0	0	0	0	12	46976	1
104	ROUTT & BLACK OAK	0	0	0	0	12	46976	1
105	ROUTT & FERNCLIFF	36555	0	1	1	12	57831	1
106	FERNCLIFF APARTMENTS N.	0	36600	0	0	12	57831	1
107	FERNCLIFF APARTMENTS S.	0	0	0	0	12	57831	1
108	FERNCLIFF &	36645	0	0	5	7	59640	1
109	FERNCLIFF &	0	0	0	0	7	59640	1
110	FERNCLIFF & HERSHBERGER	0	0	0	0	7	59640	1
111	HERSHBERGER & SWARTHMORE	0	0	0	0	7	59640	1
112	HERSHBERGER & COVE ROAD	36765	0	1	0	8	62261	1
113	COVE ROAD & GUILDHALL	0	0	0	0	8	62261	1
114	COVE ROAD & GOLFSIDE	0	0	0	0	8	62261	1
115	COVE ROAD & FRESNO	0	0	0	0	8	62261	1
116	COVE ROAD & ABBOTT	0	0	0	0	8	62261	1
117	COVE ROAD & WELLSLEY	0	0	0	0	8	62261	1
118	COVE ROAD & ASPEN	0	0	0	0	8	62261	1
119	COVE ROAD & FAIRLAND	0	0	0	0	8	62261	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	8	62261	1
121	LAFAYETTE &	36930	0	1	0	9	66436	1
122	LAFAYETTE & FLORIDA	0	0	0	0	9	66436	1
123	LAFAYETTE & MASSACHUSSETS	36975	0	2	0	11	67248	1
124	LAFAYETTE & DELAWARE	37020	0	2	0	13	67665	1
125	LAFAYETTE & CLIFTON	0	0	0	0	13	67665	1
126	LAFAYETTE & NEW YORK	37095	0	2	0	15	68314	1
127	LAFAYETTE & STAUNTON	0	0	0	0	15	68314	1
128	STAUNTON & 24 th	37140	0	2	0	17	69358	1
129	STAUNTON & 23 rd	37170	0	2	0	19	69798	1
130	STAUNTON & 22 nd	0	0	0	0	19	69798	1
131	STAUNTON & 21 st	0	0	0	0	19	69798	1
132	STAUNTON & 20 th	0	0	0	0	19	69798	1
133	STAUNTON & 19 th	0	0	0	0	19	69798	1
134	STAUNTON & 18 th	0	0	0	0	19	69798	1
135	STAUNTON & 17 th	0	0	0	0	19	69798	1
136	STAUNTON & 16 th	0	0	0	0	19	69798	1
137	STAUNTON & 15 th	0	0	0	0	19	69798	1
138	STAUNTON & EUREKA CIRCLE	0	0	0	0	19	69798	1
139	STAUNTON & 14 th	0	0	0	0	19	69798	1
140	STAUNTON & 13 th	0	0	0	0	19	69798	1
141	STAUNTON & 12 th	0	0	0	0	19	69798	1
142	STAUNTON & 11 th	0	0	0	0	19	69798	1
143	11 th & HANOVER	37365	0	2	0	21	75643	1
144	ORANGE & 10 th	0	37320	0	0	21	75643	1
145	ORANGE &	0	0	0	0	21	75643	1
146	ORANGE & 8 th	0	0	0	0	21	75643	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	21	75643	1
148	ORANGE & 5 th	0	0	0	0	21	75643	1
149	ORANGE & QUARRY	0	0	0	0	21	75643	1
150	ORANGE & GAINSBORO	0	0	0	0	21	75643	1
151	GAINSBORO & McDOWELL	0	0	0	0	21	75643	1
152	GAINSBORO & MADISON	0	0	0	0	21	75643	1

153	GAINSBORO & HARRISON	0	0	0	0	21	75643	1
154	GAINSBORO & PATTON	0	0	0	0	21	75643	1
155	1 st & GILMER	0	0	0	0	21	75643	1
156	GILMER & JEFFERSON	37650	0	1	0	22	82601	1
157	JEFFERSON & WELLS	0	0	0	0	22	82601	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	22	82601	1
159	WELLS & HOTEL ROANOKE 2	0	0	0	0	22	82601	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	22	82601	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	22	82601	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	22	82601	1
163	SALEM & 1 st	37755	0	0	1	21	85245	1
164	SALEM & JEFFERSON	37845	0	1	4	18	85906	1
165	JEFFERSON & CAMPBELL	38070	0	3	0	21	86381	1
166	JEFFERSON & CHURCH	0	38100	0	0	21	86381	1
167	JEFFERSON & FRANKLIN RD.	0	0	0	0	21	86381	1
168	JEFFERSON & BULLIT	0	0	0	0	21	86381	1
169	JEFFERSON & ELM	0	0	0	0	21	86381	1
170	JEFFERSON & MOUNTAIN	38205	0	1	0	22	88074	1
171	JEFFERSON & HIGHLAND	0	0	0	0	22	88074	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	22	88074	1
173	JEFFERSON & MAPLE	0	0	0	0	22	88074	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	22	88074	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	22	88074	1
176	WALNUT & BELLEVIEW	0	0	0	0	22	88074	1
177	BELLEVIEW & LAUREL	0	0	0	0	22	88074	1
178	BELLEVIEW & PINK	0	0	0	0	22	88074	1
179	BELLEVIEW & LINDEN	0	0	0	0	22	88074	1
180	BELLEVIEW & ASH	0	0	0	0	22	88074	1
181	ROANOKE MEMORIAL HOSPITAL	0	38520	0	0	22	88074	1
182	JEFFERSON & WELLER	0	0	0	0	22	88074	1
183	JEFFERSON & MC CLANAHAN	0	0	0	0	22	88074	1
184	JEFFERSON & 22 nd	0	0	0	0	22	88074	1
185	JEFFERSON & 23 rd	0	0	0	0	22	88074	1
186	JEFFERSON & 24 th	0	0	0	0	22	88074	1
187	JEFFERSON & 25 th	0	0	0	0	22	88074	1
188	JEFFERSON & 26 th	0	0	0	0	22	88074	1
189	26 th & RICHELIEU	0	0	0	0	22	88074	1
190	26 th & CRYSTAL SPRING	0	0	0	0	22	88074	1
191	26 th & ROSALIND	0	0	0	0	22	88074	1
192	26 th & CAROLINA	0	0	0	0	22	88074	1
193	26 th & WYCLIFF	0	0	0	0	22	88074	1
194	26 th & AVENHAM	0	0	0	0	22	88074	1
195	AVENHAM & 27 th	0	0	0	0	22	88074	1
196	AVENHAM & 28 th	0	0	0	0	22	88074	1
197	AVENHAM & 29 th	0	0	0	0	22	88074	1
198	AVENHAM & WILDWOOD	0	0	0	0	22	88074	1
199	AVENHAM & CASSELL	0	0	0	0	22	88074	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	22	88074	1
201	AVENHAM &	0	0	0	0	22	88074	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	22	88074	1
203	AVENHAM & FRANKLIN ROAD	0	0	0	0	22	88074	1

204	FRANKLIN RD. & DUKE OF GL	38820	0	1	0	23	105655	1
205	FRANKLIN RD. & K-MART	0	0	0	0	23	105655	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	23	105655	1
207	FRANKLIN RD. & PENARTH	0	0	0	0	23	105655	1
208	SUPER X	39015	0	0	2	21	110062	1
209	PENNEY'S	39105	0	1	6	16	110526	1
210		1	2			885		
1	17 12 24 85	2						
1	BRENDLES (Time Point)	39555	39300	1	5	12	110549	1
2	OGDEN & FRANKLIN ROAD	0	0	0	0	12	110549	1
3	FRANKLIN RD. & ELM VIEW R	0	0	0	0	12	110549	1
4	FRANKLIN RD. & AVENHAM MA	0	0	0	0	12	110549	1
5	FRANKLIN RD. & TANGLEWOOD	0	0	0	0	12	110549	1
6	FRANKLIN RD. & PENARTH 1	0	0	0	0	12	110549	1
7	FRANKLIN RD. & PENARTH 2	0	0	0	0	12	110549	1
8	FRANKLIN RD. & TOWNSIDE	0	0	0	0	12	110549	1
9	FRANKLIN RD. & K-MART	0	0	0	0	12	110549	1
10	FRANKLIN RD. & DUKE OF GL	0	0	0	0	12	110549	1
11	FRANKLIN RD. & WILLOW OAK	0	0	0	0	12	110549	1
12	AVENHAM & DILLARD	0	0	0	0	12	110549	1
13	AVENHAM & WHITE OAK (2)	0	0	0	0	12	110549	1
14	AVENHAM &	0	0	0	0	12	110549	1
15	AVENHAM & CLYDESDALE	0	0	0	0	12	110549	1
16	AVENHAM & CASSELL	0	0	0	0	12	110549	1
17	AVENHAM & AUDUBON	0	0	0	0	12	110549	1
18	AVENHAM & 29 th	0	0	0	0	12	110549	1
19	AVENHAM & 28 th	0	0	0	0	12	110549	1
20	AVENHAM & 27 th	0	0	0	0	12	110549	1
21	AVENHAM & 26 th	0	0	0	0	12	110549	1
22	26 th & WYCLIFFE	0	0	0	0	12	110549	1
23	26 th & CAROLINA	0	0	0	0	12	110549	1
24	26 th & ROSALIND	0	0	0	0	12	110549	1
25	26 th & CRYSTAL SPRING	0	0	0	0	12	110549	1
26	26 th & RICHELIEU	0	0	0	0	12	110549	1
27	26 th & JEFFERSON	0	0	0	0	12	110549	1
28	JEFFERSON & 25 th	0	0	0	0	12	110549	1
29	JEFFERSON & 24 th	0	0	0	0	12	110549	1
30	JEFFERSON & 23 rd	0	0	0	0	12	110549	1
31	JEFFERSON & 22 nd	0	0	0	0	12	110549	1
32	JEFFERSON & YELLOW MT.	40140	0	2	0	14	126589	1
33	JEFFERSON & WELLER	0	0	0	0	14	126589	1
34	ROANOKE MEMORIAL HOSPITAL	40185	40200	1	0	15	127818	1
35	BELLEVIEW & ASH	0	0	0	0	15	127818	1
36	BELLEVIEW & THYME	0	0	0	0	15	127818	1
37	BELLEVIEW & LINDEN	0	0	0	0	15	127818	1
38	BELLEVIEW & LAUREL	0	0	0	0	15	127818	1
39	BELLEVIEW & WALNUT	0	0	0	0	15	127818	1
40	WALNUT AVENUE BRIDGE 2	0	0	0	0	15	127818	1
41	WALNUT AVENUE BRIDGE 1	0	0	0	0	15	127818	1
42	JEFFERSON & MAPLE	0	0	0	0	15	127818	1
43	JEFFERSON & ALBEMARLE	0	0	0	0	15	127818	1

44	JEFFERSON & HIGHLAND	0	0	0	0	15	127818	1
45	JEFFERSON & MOUNTAIN	0	0	0	0	15	127818	1
46	JEFFERSON & ELM	0	0	0	0	15	127818	1
47	JEFFERSON & BULLIT	0	0	0	0	15	127818	1
48	JEFFERSON & FRANKLIN RD.	0	0	0	0	15	127818	1
49	JEFFERSON & CHURCH	40920	40800	2	4	13	135658	1
50	JEFFERSON & CAMPBELL	40995	0	3	0	16	136098	1
51	CAMPBELL & MARKET ST.	41100	0	3	0	19	136701	1
52	WILLIAMSON ROAD & SHENAND	0	0	0	0	19	136701	1
53	WELLS & COMMONWEALTH	0	0	0	0	19	136701	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	19	136701	1
55	WELLS & JEFFERSON	0	0	0	0	19	136701	1
56	JEFFERSON & GILMER	0	0	0	0	19	136701	1
57	GILMER & GAINSBORO	0	0	0	0	19	136701	1
58	GAINSBORO & PATTON	41265	0	1	0	20	140110	1
59	GAINSBORO & HARRISON	0	0	0	0	20	140110	1
60	GAINSBORO & MADISON	0	0	0	0	20	140110	1
61	GAINSBORO & McDOWELL	0	0	0	0	20	140110	1
62	GAINSBORO & ORANGE	0	0	0	0	20	140110	1
63	ORANGE & QUARRY	0	0	0	0	20	140110	1
64	ORANGE & 5 th	0	0	0	0	20	140110	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	20	140110	1
66	ORANGE & 8 th	0	0	0	0	20	140110	1
67	ORANGE & 10 th	0	41280	0	0	20	140110	1
68	ORANGE & 11 th	0	0	0	0	20	140110	1
69	11 th & HANNOVER	0	0	0	0	20	140110	1
70	11 th & STAUNTON	41550	0	0	1	19	146837	1
71	STAUNTON & 12 th	0	0	0	0	19	146837	1
72	STAUNTON & 13 th	0	0	0	0	19	146837	1
73	STAUNTON & 14 th	0	0	0	0	19	146837	1
74	STAUNTON & EUREKA CIRCLE	0	0	0	0	19	146837	1
75	STAUNTON & 15 th	0	0	0	0	19	146837	1
76	STAUNTON & 16 th	0	0	0	0	19	146837	1
77	STAUNTON & 17 th	0	0	0	0	19	146837	1
78	STAUNTON & 18 th	41625	0	0	1	18	149945	1
79	STAUNTON & 19 th	0	0	0	0	18	149945	1
80	STAUNTON & 20 th	0	0	0	0	18	149945	1
81	STAUNTON & 21 st	0	0	0	0	18	149945	1
82	STAUNTON & 22 nd	0	0	0	0	18	149945	1
83	STAUNTON & 23 rd	0	0	0	0	18	149945	1
84	STAUNTON & 24 th	0	0	0	0	18	149945	1
85	STAUNTON & LAFAYETTE	0	0	0	0	18	149945	1
86	LAFAYETTE & NEW YORK	0	0	0	0	18	149945	1
87	LAFAYETTE & CLIFTON	0	0	0	0	18	149945	1
88	LAFAYETTE & DELAWARE	0	0	0	0	18	149945	1
89	LAFAYETTE & MASSACHUSSETS	0	0	0	0	18	149945	1
90	LAFAYETTE & FLORIDA	0	0	0	0	18	149945	1
91	LAFAYETTE &	0	0	0	0	18	149945	1
92	LAFAYETTE & COVE ROAD	0	0	0	0	18	149945	1
93	COVE ROAD & FAIRLAND	0	0	0	0	18	149945	1
94	COVE ROAD & ASPEN	0	0	0	0	18	149945	1

95	COVE ROAD & WELLSLEY	0	0	0	0	18	149945	1
96	COVE ROAD & ABBOTT	0	0	0	0	18	149945	1
97	COVE ROAD & COVELAND	0	0	0	0	18	149945	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	18	149945	1
99	COVE ROAD & GUILDHALL	0	0	0	0	18	149945	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	18	149945	1
101	COVE ROAD &	0	0	0	0	18	149945	1
102	COVE ROAD &	0	0	0	0	18	149945	1
103	COVE ROAD & ROUTT	0	0	0	0	18	149945	1
104	ROUTT & BLACK OAK	0	0	0	0	18	149945	1
105	ROUTT & FERNCLIFF	0	0	0	0	18	149945	1
106	FERNCLIFF APARTMENTS N.	42285	42000	1	1	18	164651	1
107	FERNCLIFF APARTMENTS S.	0	0	0	0	18	164651	1
108	FERNCLIFF &	0	0	0	0	18	164651	1
109	FERNCLIFF &	42390	0	5	2	21	166460	1
110	FERNCLIFF & HERSHBERGER	0	0	0	0	21	166460	1
111	HERSHBERGER & SWARTHMORE	0	0	0	0	21	166460	1
112	HERSHBERGER & COVE ROAD	42480	0	2	0	23	168548	1
113	COVE ROAD & GUILDHALL	0	0	0	0	23	168548	1
114	COVE ROAD & GOLFSIDE	0	0	0	0	23	168548	1
115	COVE ROAD & FRESNO	0	0	0	0	23	168548	1
116	COVE ROAD & ABBOTT	0	0	0	0	23	168548	1
117	COVE ROAD & WELLSLEY	0	0	0	0	23	168548	1
118	COVE ROAD & ASPEN	0	0	0	0	23	168548	1
119	COVE ROAD & FAIRLAND	0	0	0	0	23	168548	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	23	168548	1
121	LAFAYETTE &	0	0	0	0	23	168548	1
122	LAFAYETTE & FLORIDA	0	0	0	0	23	168548	1
123	LAFAYETTE & MASSACHUSSETS	42615	0	0	1	22	174092	1
124	LAFAYETTE & DELAWARE	0	0	0	0	22	174092	1
125	LAFAYETTE & CLIFTON	0	0	0	0	22	174092	1
126	LAFAYETTE & NEW YORK	0	0	0	0	22	174092	1
127	LAFAYETTE & STAUNTON	0	0	0	0	22	174092	1
128	STAUNTON & 24 th	0	0	0	0	22	174092	1
129	STAUNTON & 23 rd	42705	0	0	1	21	176690	1
130	STAUNTON & 22 nd	0	0	0	0	21	176690	1
131	STAUNTON & 21 st	0	0	0	0	21	176690	1
132	STAUNTON & 20 th	0	0	0	0	21	176690	1
133	STAUNTON & 19 th	0	0	0	0	21	176690	1
134	STAUNTON & 18 th	0	0	0	0	21	176690	1
135	STAUNTON & 17 th	0	0	0	0	21	176690	1
136	STAUNTON & 16 th	0	0	0	0	21	176690	1
137	STAUNTON & 15 th	0	0	0	0	21	176690	1
138	STAUNTON & EUREKA CIRCLE	0	0	0	0	21	176690	1
139	STAUNTON & 14 th	0	0	0	0	21	176690	1
140	STAUNTON & 13 th	0	0	0	0	21	176690	1
141	STAUNTON & 12 th	0	0	0	0	21	176690	1
142	STAUNTON & 11 th	0	0	0	0	21	176690	1
143	11 th & HANOVER	0	0	0	0	21	176690	1
144	ORANGE & 10 th	0	43020	0	0	21	176690	1
145	ORANGE &	0	0	0	0	21	176690	1

146	ORANGE & 8 th	42990	0	1	0	22	184089	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	22	184089	1
148	ORANGE & 5 th	0	0	0	0	22	184089	1
149	ORANGE & QUARRY	0	0	0	0	22	184089	1
150	ORANGE & GAINSBORO	0	0	0	0	22	184089	1
151	GAINSBORO & McDOWELL	0	0	0	0	22	184089	1
152	GAINSBORO & MADISON	0	0	0	0	22	184089	1
153	GAINSBORO & HARRISON	0	0	0	0	22	184089	1
154	GAINSBORO & PATTON	43125	0	0	2	20	188867	1
155	1 st & GILMER	0	0	0	0	20	188867	1
156	GILMER & JEFFERSON	0	0	0	0	20	188867	1
157	JEFFERSON & WELLS	43170	0	0	1	19	189702	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	19	189702	1
159	WELLS & HOTEL ROANOKE 2	0	0	0	0	19	189702	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	19	189702	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	19	189702	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	19	189702	1
163	SALEM & 1 st	43350	0	0	1	18	191998	1
164	SALEM & JEFFERSON	0	0	0	0	18	191998	1
165	JEFFERSON & CAMPBELL	43605	0	11	0	29	193134	1
166	JEFFERSON & CHURCH	43710	43800	1	0	30	193180	1
167	JEFFERSON & FRANKLIN RD.	0	0	0	0	30	193180	1
168	JEFFERSON & BULLIT	0	0	0	0	30	193180	1
169	JEFFERSON & ELM	0	0	0	0	30	193180	1
170	JEFFERSON & MOUNTAIN	43815	0	3	0	33	194873	1
171	JEFFERSON & HIGHLAND	0	0	0	0	33	194873	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	33	194873	1
173	JEFFERSON & MAPLE	0	0	0	0	33	194873	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	33	194873	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	33	194873	1
176	WALNUT & BELLEVIEW	0	0	0	0	33	194873	1
177	BELLEVIEW & LAUREL	0	0	0	0	33	194873	1
178	BELLEVIEW & PINK	0	0	0	0	33	194873	1
179	BELLEVIEW & LINDEN	0	0	0	0	33	194873	1
180	BELLEVIEW & ASH	0	0	0	0	33	194873	1
181	ROANOKE MEMORIAL HOSPITAL	0	44220	0	0	33	194873	1
182	JEFFERSON & WELLER	0	0	0	0	33	194873	1
183	JEFFERSON & MC CLANAHAN	44070	0	1	0	34	202365	1
184	JEFFERSON & 22 nd	44100	0	0	1	33	202898	1
185	JEFFERSON & 23 rd	0	0	0	0	33	202898	1
186	JEFFERSON & 24 th	44145	0	1	0	34	203802	1
187	JEFFERSON & 25 th	0	0	0	0	34	203802	1
188	JEFFERSON & 26 th	0	0	0	0	34	203802	1
189	26 th & RICHELIEU	0	0	0	0	34	203802	1
190	26 th & CRYSTAL SPRING	0	0	0	0	34	203802	1
191	26 th & ROSALIND	0	0	0	0	34	203802	1
192	26 th & CAROLINA	0	0	0	0	34	203802	1
193	26 th & WYCLIFF	0	0	0	0	34	203802	1
194	26 th & AVENHAM	0	0	0	0	34	203802	1
195	AVENHAM & 27 th	0	0	0	0	34	203802	1
196	AVENHAM & 28 th	0	0	0	0	34	203802	1

197	AVENHAM & 29 th	0	0	0	0	34	203802	1
198	AVENHAM & WILDWOOD	0	0	0	0	34	203802	1
199	AVENHAM & CASSELL	0	0	0	0	34	203802	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	34	203802	1
201	AVENHAM &	0	0	0	0	34	203802	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	34	203802	1
203	AVENHAM & FRANKLIN ROAD	0	0	0	0	34	203802	1
204	FRANKLIN RD. & DUKE OF GL	0	0	0	0	34	203802	1
205	FRANKLIN RD. & K-MART	0	0	0	0	34	203802	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	34	203802	1
207	FRANKLIN RD. & PENARTH	44550	0	0	1	33	214008	1
208	SUPER X	44685	0	2	1	34	216792	1
209	PENNEY'S	44760	0	0	6	28	217279	1
210		1	2			2205		
1	17 12 24 85	2						
1	BRENDLES (Time Point)	45060	45000	6	1	33	217302	1
2	OGDEN & FRANKLIN ROAD	0	0	0	0	33	217302	1
3	FRANKLIN RD. & ELM VIEW R	0	0	0	0	33	217302	1
4	FRANKLIN RD. & AVENHAM MA	0	0	0	0	33	217302	1
5	FRANKLIN RD. & TANGLEWOOD	0	0	0	0	33	217302	1
6	FRANKLIN RD. & PENARTH 1	0	0	0	0	33	217302	1
7	FRANKLIN RD. & PENARTH 2	0	0	0	0	33	217302	1
8	FRANKLIN RD. & TOWNSIDE	0	0	0	0	33	217302	1
9	FRANKLIN RD. & K-MART	0	0	0	0	33	217302	1
10	FRANKLIN RD. & DUKE OF GL	0	0	0	0	33	217302	1
11	FRANKLIN RD. & WILLOW OAK	0	0	0	0	33	217302	1
12	AVENHAM & DILLARD	0	0	0	0	33	217302	1
13	AVENHAM & WHITE OAK (2)	45405	0	0	1	32	225026	1
14	AVENHAM &	0	0	0	0	32	225026	1
15	AVENHAM & CLYDESDALE	45450	0	1	0	33	226348	1
16	AVENHAM & CASSELL	0	0	0	0	33	226348	1
17	AVENHAM & AUDUBON	45495	0	1	0	34	227206	1
18	AVENHAM & 29 th	0	0	0	0	34	227206	1
19	AVENHAM & 28 th	0	0	0	0	34	227206	1
20	AVENHAM & 27 th	0	0	0	0	34	227206	1
21	AVENHAM & 26 th	0	0	0	0	34	227206	1
22	26 th & WYCLIFFE	45570	0	1	0	35	229502	1
23	26 th & CAROLINA	0	0	0	0	35	229502	1
24	26 th & ROSALIND	0	0	0	0	35	229502	1
25	26 th & CRYSTAL SPRING	0	0	0	0	35	229502	1
26	26 th & RICHELIEU	0	0	0	0	35	229502	1
27	26 th & JEFFERSON	0	0	0	0	35	229502	1
28	JEFFERSON & 25 th	0	0	0	0	35	229502	1
29	JEFFERSON & 24 th	0	0	0	0	35	229502	1
30	JEFFERSON & 23 rd	0	0	0	0	35	229502	1
31	JEFFERSON & 22 nd	0	0	0	0	35	229502	1
32	JEFFERSON & YELLOW MT.	45690	0	1	0	36	233538	1
33	JEFFERSON & WELLER	0	0	0	0	36	233538	1
34	ROANOKE MEMORIAL HOSPITAL	45900	45900	2	0	38	234744	1
35	BELLEVIEW & ASH	0	0	0	0	38	234744	1
36	BELLEVIEW & THYME	0	0	0	0	38	234744	1

37	BELLEVIEW & LINDEN	0	0	0	0	38	234744	1
38	BELLEVIEW & LAUREL	0	0	0	0	38	234744	1
39	BELLEVIEW & WALNUT	0	0	0	0	38	234744	1
40	WALNUT AVENUE BRIDGE 2	0	0	0	0	38	234744	1
41	WALNUT AVENUE BRIDGE 1	0	0	0	0	38	234744	1
42	JEFFERSON & MAPLE	46140	0	0	1	37	239708	1
43	JEFFERSON & ALBEMARLE	0	0	0	0	37	239708	1
44	JEFFERSON & HIGHLAND	0	0	0	0	37	239708	1
45	JEFFERSON & MOUNTAIN	0	0	0	0	37	239708	1
46	JEFFERSON & ELM	0	0	0	0	37	239708	1
47	JEFFERSON & BULLIT	0	0	0	0	37	239708	1
48	JEFFERSON & FRANKLIN RD.	0	0	0	0	37	239708	1
49	JEFFERSON & CHURCH	46425	46500	3	5	35	242631	1
50	JEFFERSON & CAMPBELL	0	0	0	0	35	242631	1
51	CAMPBELL & MARKET ST.	46635	0	7	0	42	243698	1
52	WILLIAMSON ROAD & SHENAND	0	0	0	0	42	243698	1
53	WELLS & COMMONWEALTH	0	0	0	0	42	243698	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	42	243698	1
55	WELLS & JEFFERSON	0	0	0	0	42	243698	1
56	JEFFERSON & GILMER	0	0	0	0	42	243698	1
57	GILMER & GAINSBORO	0	0	0	0	42	243698	1
58	GAINSBORO & PATTON	0	0	0	0	42	243698	1
59	GAINSBORO & HARRISON	0	0	0	0	42	243698	1
60	GAINSBORO & MADISON	0	0	0	0	42	243698	1
61	GAINSBORO & McDOWELL	0	0	0	0	42	243698	1
62	GAINSBORO & ORANGE	0	0	0	0	42	243698	1
63	ORANGE & QUARRY	0	0	0	0	42	243698	1
64	ORANGE & 5 th	0	0	0	0	42	243698	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	42	243698	1
66	ORANGE & 8 th	0	0	0	0	42	243698	1
67	ORANGE & 10 th	46995	46980	1	0	43	252606	1
68	ORANGE & 11 th	47025	0	0	1	42	253278	1
69	11 th & HANNOVER	0	0	0	0	42	253278	1
70	11 th & STAUNTON	0	0	0	0	42	253278	1
71	STAUNTON & 12 th	0	0	0	0	42	253278	1
72	STAUNTON & 13 th	47100	0	1	0	43	254670	1
73	STAUNTON & 14 th	47130	0	0	1	42	254948	1
74	STAUNTON & EUREKA CIRCLE	0	0	0	0	42	254948	1
75	STAUNTON & 15 th	0	0	0	0	42	254948	1
76	STAUNTON & 16 th	0	0	0	0	42	254948	1
77	STAUNTON & 17 th	0	0	0	0	42	254948	1
78	STAUNTON & 18 th	0	0	0	0	42	254948	1
79	STAUNTON & 19 th	0	0	0	0	42	254948	1
80	STAUNTON & 20 th	0	0	0	0	42	254948	1
81	STAUNTON & 21 st	0	0	0	0	42	254948	1
82	STAUNTON & 22 nd	0	0	0	0	42	254948	1
83	STAUNTON & 23 rd	47265	0	0	2	40	259169	1
84	STAUNTON & 24 th	0	0	0	0	40	259169	1
85	STAUNTON & LAFAYETTE	0	0	0	0	40	259169	1
86	LAFAYETTE & NEW YORK	0	0	0	0	40	259169	1
87	LAFAYETTE & CLIFTON	0	0	0	0	40	259169	1

88	LAFAYETTE & DELAWARE	47370	0	0	2	38	261373	1
89	LAFAYETTE & MASSACHUSSETS	0	0	0	0	38	261373	1
90	LAFAYETTE & FLORIDA	0	0	0	0	38	261373	1
91	LAFAYETTE &	0	0	0	0	38	261373	1
92	LAFAYETTE & COVE ROAD	0	0	0	0	38	261373	1
93	COVE ROAD & FAIRLAND	0	0	0	0	38	261373	1
94	COVE ROAD & ASPEN	0	0	0	0	38	261373	1
95	COVE ROAD & WELLSLEY	0	0	0	0	38	261373	1
96	COVE ROAD & ABBOTT	0	0	0	0	38	261373	1
97	COVE ROAD & COVELAND	47505	0	0	1	37	265757	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	37	265757	1
99	COVE ROAD & GUILDHALL	0	0	0	0	37	265757	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	37	265757	1
101	COVE ROAD &	0	0	0	0	37	265757	1
102	COVE ROAD &	0	0	0	0	37	265757	1
103	COVE ROAD & ROUTT	0	0	0	0	37	265757	1
104	ROUTT & BLACK OAK	0	0	0	0	37	265757	1
105	ROUTT & FERNCLIFF	0	0	0	0	37	265757	1
106	FERNCLIFF APARTMENTS N.	47730	47700	1	1	37	271579	1
107	FERNCLIFF APARTMENTS S.	0	0	0	0	37	271579	1
108	FERNCLIFF &	47775	0	0	1	36	272994	1
109	FERNCLIFF &	0	0	0	0	36	272994	1
110	FERNCLIFF & HERSHBERGER	0	0	0	0	36	272994	1
111	HERSHBERGER & SWARTHMORE	0	0	0	0	36	272994	1
112	HERSHBERGER & COVE ROAD	0	0	0	0	36	272994	1
113	COVE ROAD & GUILDHALL	47895	0	1	0	37	276032	1
114	COVE ROAD & GOLFSIDE	0	0	0	0	37	276032	1
115	COVE ROAD & FRESNO	0	0	0	0	37	276032	1
116	COVE ROAD & ABBOTT	0	0	0	0	37	276032	1
117	COVE ROAD & WELLSLEY	47955	0	1	0	38	277841	1
118	COVE ROAD & ASPEN	0	0	0	0	38	277841	1
119	COVE ROAD & FAIRLAND	0	0	0	0	38	277841	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	38	277841	1
121	LAFAYETTE &	0	0	0	0	38	277841	1
122	LAFAYETTE & FLORIDA	48030	0	1	0	39	280184	1
123	LAFAYETTE & MASSACHUSSETS	48060	0	1	0	40	280972	1
124	LAFAYETTE & DELAWARE	0	0	0	0	40	280972	1
125	LAFAYETTE & CLIFTON	0	0	0	0	40	280972	1
126	LAFAYETTE & NEW YORK	0	0	0	0	40	280972	1
127	LAFAYETTE & STAUNTON	0	0	0	0	40	280972	1
128	STAUNTON & 24 th	0	0	0	0	40	280972	1
129	STAUNTON & 23 rd	48165	0	1	0	41	283570	1
130	STAUNTON & 22 nd	0	0	0	0	41	283570	1
131	STAUNTON & 21 st	0	0	0	0	41	283570	1
132	STAUNTON & 20 th	48210	0	0	1	40	284915	1
133	STAUNTON & 19 th	0	0	0	0	40	284915	1
134	STAUNTON & 18 th	0	0	0	0	40	284915	1
135	STAUNTON & 17 th	0	0	0	0	40	284915	1
136	STAUNTON & 16 th	0	0	0	0	40	284915	1
137	STAUNTON & 15 th	0	0	0	0	40	284915	1
138	STAUNTON & EUREKA CIRCLE	0	0	0	0	40	284915	1

139	STAUNTON & 14 th	0	0	0	0	40	284915	1
140	STAUNTON & 13 th	0	0	0	0	40	284915	1
141	STAUNTON & 12 th	0	0	0	0	40	284915	1
142	STAUNTON & 11 th	0	0	0	0	40	284915	1
143	11 th & HANOVER	48360	0	3	0	43	289392	1
144	ORANGE & 10 th	0	48420	0	0	43	289392	1
145	ORANGE &	0	0	0	0	43	289392	1
146	ORANGE & 8 th	0	0	0	0	43	289392	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	43	289392	1
148	ORANGE & 5 th	0	0	0	0	43	289392	1
149	ORANGE & QUARRY	0	0	0	0	43	289392	1
150	ORANGE & GAINSBORO	0	0	0	0	43	289392	1
151	GAINSBORO & McDOWELL	0	0	0	0	43	289392	1
152	GAINSBORO & MADISON	0	0	0	0	43	289392	1
153	GAINSBORO & HARRISON	0	0	0	0	43	289392	1
154	GAINSBORO & PATTON	48585	0	0	1	42	295748	1
155	1 st & GILMER	0	0	0	0	42	295748	1
156	GILMER & JEFFERSON	0	0	0	0	42	295748	1
157	JEFFERSON & WELLS	0	0	0	0	42	295748	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	42	295748	1
159	WELLS & HOTEL ROANOKE 2	0	0	0	0	42	295748	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	42	295748	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	42	295748	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	42	295748	1
163	SALEM & 1 st	48750	0	1	0	43	298972	1
164	SALEM & JEFFERSON	48810	0	0	1	42	299575	1
165	JEFFERSON & CAMPBELL	48915	0	6	0	48	300038	1
166	JEFFERSON & CHURCH	49155	49200	4	2	50	300061	1
167	JEFFERSON & FRANKLIN RD.	49185	0	2	0	52	300223	1
168	JEFFERSON & BULLIT	0	0	0	0	52	300223	1
169	JEFFERSON & ELM	49260	0	0	1	51	301174	1
170	JEFFERSON & MOUNTAIN	0	0	0	0	51	301174	1
171	JEFFERSON & HIGHLAND	0	0	0	0	51	301174	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	51	301174	1
173	JEFFERSON & MAPLE	0	0	0	0	51	301174	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	51	301174	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	51	301174	1
176	WALNUT & BELLEVIEW	0	0	0	0	51	301174	1
177	BELLEVIEW & LAUREL	0	0	0	0	51	301174	1
178	BELLEVIEW & PINK	0	0	0	0	51	301174	1
179	BELLEVIEW & LINDEN	0	0	0	0	51	301174	1
180	BELLEVIEW & ASH	0	0	0	0	51	301174	1
181	ROANOKE MEMORIAL HOSPITAL	49530	49620	0	1	50	307947	1
182	JEFFERSON & WELLER	0	0	0	0	50	307947	1
183	JEFFERSON & MC CLANAHAN	0	0	0	0	50	307947	1
184	JEFFERSON & 22 nd	49620	0	1	1	50	309756	1
185	JEFFERSON & 23 rd	0	0	0	0	50	309756	1
186	JEFFERSON & 24 th	0	0	0	0	50	309756	1
187	JEFFERSON & 25 th	0	0	0	0	50	309756	1
188	JEFFERSON & 26 th	0	0	0	0	50	309756	1
189	26 th & RICHELIEU	0	0	0	0	50	309756	1

190	26 th & CRYSTAL SPRING	0	0	0	0	50	309756	1
191	26 th & ROSALIND	0	0	0	0	50	309756	1
192	26 th & CAROLINA	0	0	0	0	50	309756	1
193	26 th & WYCLIFF	0	0	0	0	50	309756	1
194	26 th & AVENHAM	49800	0	0	1	49	313815	1
195	AVENHAM & 27 th	0	0	0	0	49	313815	1
196	AVENHAM & 28 th	0	0	0	0	49	313815	1
197	AVENHAM & 29 th	0	0	0	0	49	313815	1
198	AVENHAM & WILDWOOD	0	0	0	0	49	313815	1
199	AVENHAM & CASSELL	0	0	0	0	49	313815	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	49	313815	1
201	AVENHAM &	0	0	0	0	49	313815	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	49	313815	1
203	AVENHAM & FRANKLIN ROAD	0	0	0	0	49	313815	1
204	FRANKLIN RD. & DUKE OF GL	49995	0	0	2	47	319266	1
205	FRANKLIN RD. & K-MART	0	0	0	0	47	319266	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	47	319266	1
207	FRANKLIN RD. & PENARTH	0	0	0	0	47	319266	1
208	SUPER X	50190	0	0	2	45	323650	1
209	PENNEY'S	50310	0	6	4	47	324113	1
210		1	2			3030		
1	17 12 24 85	2						
1	BRENDLES (Time Point)	50475	50400	3	0	50	324136	1
2	OGDEN & FRANKLIN ROAD	0	0	0	0	50	324136	1
3	FRANKLIN RD. & ELM VIEW R	0	0	0	0	50	324136	1
4	FRANKLIN RD. & AVENHAM MA	0	0	0	0	50	324136	1
5	FRANKLIN RD. & TANGLEWOOD	50685	0	0	1	49	326966	1
6	FRANKLIN RD. & PENARTH 1	0	0	0	0	49	326966	1
7	FRANKLIN RD. & PENARTH 2	50835	0	1	0	50	329402	1
8	FRANKLIN RD. & TOWNSIDE	0	0	0	0	50	329402	1
9	FRANKLIN RD. & K-MART	0	0	0	0	50	329402	1
10	FRANKLIN RD. & DUKE OF GL	50880	0	0	1	49	330446	1
11	FRANKLIN RD. & WILLOW OAK	0	0	0	0	49	330446	1
12	AVENHAM & DILLARD	0	0	0	0	49	330446	1
13	AVENHAM & WHITE OAK (2)	0	0	0	0	49	330446	1
14	AVENHAM &	0	0	0	0	49	330446	1
15	AVENHAM & CLYDESDALE	0	0	0	0	49	330446	1
16	AVENHAM & CASSELL	50985	0	1	0	50	333577	1
17	AVENHAM & AUDUBON	0	0	0	0	50	333577	1
18	AVENHAM & 29 th	0	0	0	0	50	333577	1
19	AVENHAM & 28 th	0	0	0	0	50	333577	1
20	AVENHAM & 27 th	0	0	0	0	50	333577	1
21	AVENHAM & 26 th	0	0	0	0	50	333577	1
22	26 th & WYCLIFFE	0	0	0	0	50	333577	1
23	26 th & CAROLINA	0	0	0	0	50	333577	1
24	26 th & ROSALIND	0	0	0	0	50	333577	1
25	26 th & CRYSTAL SPRING	0	0	0	0	50	333577	1
26	26 th & RICHELIEU	0	0	0	0	50	333577	1
27	26 th & JEFFERSON	0	0	0	0	50	333577	1
28	JEFFERSON & 25 th	0	0	0	0	50	333577	1
29	JEFFERSON & 24 th	0	0	0	0	50	333577	1

30	JEFFERSON & 23 rd	0	0	0	0	50	333577	1
31	JEFFERSON & 22 nd	51195	0	0	1	49	340025	1
32	JEFFERSON & YELLOW MT.	51255	0	2	0	51	340373	1
33	JEFFERSON & WELLER	51300	0	1	0	52	341092	1
34	ROANOKE MEMORIAL HOSPITAL	0	51300	0	0	52	341092	1
35	BELLEVIEW & ASH	0	0	0	0	52	341092	1
36	BELLEVIEW & THYME	0	0	0	0	52	341092	1
37	BELLEVIEW & LINDEN	0	0	0	0	52	341092	1
38	BELLEVIEW & LAUREL	0	0	0	0	52	341092	1
39	BELLEVIEW & WALNUT	0	0	0	0	52	341092	1
40	WALNUT AVENUE BRIDGE 2	0	0	0	0	52	341092	1
41	WALNUT AVENUE BRIDGE 1	0	0	0	0	52	341092	1
42	JEFFERSON & MAPLE	0	0	0	0	52	341092	1
43	JEFFERSON & ALBEMARLE	0	0	0	0	52	341092	1
44	JEFFERSON & HIGHLAND	0	0	0	0	52	341092	1
45	JEFFERSON & MOUNTAIN	51555	0	0	3	49	347796	1
46	JEFFERSON & ELM	0	0	0	0	49	347796	1
47	JEFFERSON & BULLIT	0	0	0	0	49	347796	1
48	JEFFERSON & FRANKLIN RD.	0	0	0	0	49	347796	1
49	JEFFERSON & CHURCH	51765	51900	2	3	48	349420	1
50	JEFFERSON & CAMPBELL	51825	0	3	0	51	349837	1
51	CAMPBELL & MARKET ST.	51870	0	0	1	50	349883	1
52	WILLIAMSON ROAD & SHENAND	51915	0	3	0	53	350416	1
53	WELLS & COMMONWEALTH	0	0	0	0	53	350416	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	53	350416	1
55	WELLS & JEFFERSON	0	0	0	0	53	350416	1
56	JEFFERSON & GILMER	0	0	0	0	53	350416	1
57	GILMER & GAINSBORO	0	0	0	0	53	350416	1
58	GAINSBORO & PATTON	0	0	0	0	53	350416	1
59	[illegible]	5209[illegible]	0	1	0	54	352968	1
60	[illegible]	[illegible]	0	0	[illegible]	54	352968	1
61	GAINSBORO & McDOWELL	0	0	0	0	54	352968	1
62	GAINSBORO & ORANGE	0	0	0	0	54	352968	1
63	ORANGE & QUARRY	0	0	0	0	54	352968	1
64	ORANGE & 5 th	0	0	0	0	54	352968	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	54	352968	1
66	ORANGE & 8 th	0	0	0	0	54	352968	1
67	ORANGE & 10 th	52275	52380	1	0	55	358581	1
68	ORANGE & 11 th	0	0	0	0	55	358581	1
69	11 th & HANNOVER	0	0	0	0	55	358581	1
70	11 th & STAUNTON	0	0	0	0	55	358581	1
71	STAUNTON & 12 th	0	0	0	0	55	358581	1
72	STAUNTON & 13 th	0	0	0	0	55	358581	1
73	STAUNTON & 14 th	0	0	0	0	55	358581	1
74	STAUNTON & EUREKA CIRCLE	0	0	0	0	55	358581	1
75	STAUNTON & 15 th	0	0	0	0	55	358581	1
76	STAUNTON & 16 th	0	0	0	0	55	358581	1
77	STAUNTON & 17 th	0	0	0	0	55	358581	1
78	STAUNTON & 18 th	0	0	0	0	55	358581	1
79	STAUNTON & 19 th	0	0	0	0	55	358581	1
80	STAUNTON & 20 th	0	0	0	0	55	358581	1

81	STAUNTON & 21 st	0	0	0	0	55	358581	1
82	STAUNTON & 22 nd	0	0	0	0	55	358581	1
83	STAUNTON & 23 rd	0	0	0	0	55	358581	1
84	STAUNTON & 24 th	0	0	0	0	55	358581	1
85	STAUNTON & LAFAYETTE	0	0	0	0	55	358581	1
86	LAFAYETTE & NEW YORK	0	0	0	0	55	358581	1
87	LAFAYETTE & CLIFTON	0	0	0	0	55	358581	1
88	LAFAYETTE & DELAWARE	0	0	0	0	55	358581	1
89	LAFAYETTE & MASSACHUSSETS	52680	0	0	2	53	367604	1
90	LAFAYETTE & FLORIDA	52725	0	1	1	53	368207	1
91	LAFAYETTE &	0	0	0	0	53	368207	1
92	LAFAYETTE & COVE ROAD	0	0	0	0	53	368207	1
93	COVE ROAD & FAIRLAND	0	0	0	0	53	368207	1
94	COVE ROAD & ASPEN	0	0	0	0	53	368207	1
95	COVE ROAD & WELLSLEY	0	0	0	0	53	368207	1
96	COVE ROAD & ABBOTT	0	0	0	0	53	368207	1
97	COVE ROAD & COVELAND	52845	0	0	1	52	371384	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	52	371384	1
99	COVE ROAD & GUILDHALL	0	0	0	0	52	371384	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	52	371384	1
101	COVE ROAD &	52935	0	0	1	51	374144	1
102	COVE ROAD &	0	0	0	0	51	374144	1
103	COVE ROAD & ROUTT	0	0	0	0	51	374144	1
104	ROUTT & BLACK OAK	0	0	0	0	51	374144	1
105	ROUTT & FERNCLIFF	0	0	0	0	51	374144	1
106	FERNCLIFF APARTMENTS N.	0	53100	0	0	51	374144	1
107	FERNCLIFF APARTMENTS S.	0	0	0	0	51	374144	1
108	FERNCLIFF &	0	0	0	0	51	374144	1
109	FERNCLIFF &	0	0	0	0	51	374144	1
110	FERNCLIFF & HERSHBERGER	53385	0	2	0	53	379293	1
111	HERSHBERGER & SWARTHMORE	0	0	0	0	53	379293	1
112	HERSHBERGER & COVE ROAD	53445	0	3	0	56	380267	1
113	COVE ROAD & GUILDHALL	0	0	0	0	56	380267	1
114	COVE ROAD & GOLFSIDE	53520	0	0	1	55	381311	1
115	COVE ROAD & FRESNO	0	0	0	0	55	381311	1
116	COVE ROAD & ABBOTT	0	0	0	0	55	381311	1
117	COVE ROAD & WELLSLEY	0	0	0	0	55	381311	1
118	COVE ROAD & ASPEN	0	0	0	0	55	381311	1
119	COVE ROAD & FAIRLAND	0	0	0	0	55	381311	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	55	381311	1
121	LAFAYETTE &	0	0	0	0	55	381311	1
122	LAFAYETTE & FLORIDA	0	0	0	0	55	381311	1
123	LAFAYETTE & MASSACHUSSETS	0	0	0	0	55	381311	1
124	LAFAYETTE & DELAWARE	0	0	0	0	55	381311	1
125	LAFAYETTE & CLIFTON	0	0	0	0	55	381311	1
126	LAFAYETTE & NEW YORK	0	0	0	0	55	381311	1
127	LAFAYETTE & STAUNTON	0	0	0	0	55	381311	1
128	STAUNTON & 24 th	0	0	0	0	55	381311	1
129	STAUNTON & 23 rd	0	0	0	0	55	381311	1
130	STAUNTON & 22 nd	0	0	0	0	55	381311	1
131	STAUNTON & 21 st	0	0	0	0	55	381311	1

132	STAUNTON & 20 th	53880	0	1	0	56	390033	1
133	STAUNTON & 19 th	0	0	0	0	56	390033	1
134	STAUNTON & 18 th	0	0	0	0	56	390033	1
135	STAUNTON & 17 th	0	0	0	0	56	390033	1
136	STAUNTON & 16 th	0	0	0	0	56	390033	1
137	STAUNTON & 15 th	0	0	0	0	56	390033	1
138	STAUNTON & EUREKA CIRCLE	0	0	0	0	56	390033	1
139	STAUNTON & 14 th	0	0	0	0	56	390033	1
140	STAUNTON & 13 th	0	0	0	0	56	390033	1
141	STAUNTON & 12 th	0	0	0	0	56	390033	1
142	STAUNTON & 11 th	0	0	0	0	56	390033	1
143	11 th & HANOVER	0	0	0	0	56	390033	1
144	ORANGE & 10 th	0	54120	0	0	56	390033	1
145	ORANGE &	0	0	0	0	56	390033	1
146	ORANGE & 8 th	54090	0	1	0	57	396272	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	57	396272	1
148	ORANGE & 5 th	54240	0	0	2	55	397872	1
149	ORANGE & QUARRY	0	0	0	0	55	397872	1
150	ORANGE & GAINSBORO	0	0	0	0	55	397872	1
151	GAINSBORO & McDOWELL	0	0	0	0	55	397872	1
152	GAINSBORO & MADISON	0	0	0	0	55	397872	1
153	GAINSBORO & HARRISON	54345	0	1	0	56	400795	1
154	GAINSBORO & PATTON	0	0	0	0	56	400795	1
155	1 st & GILMER	0	0	0	0	56	400795	1
156	GILMER & JEFFERSON	0	0	0	0	56	400795	1
157	JEFFERSON & WELLS	0	0	0	0	56	400795	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	56	400795	1
159	WELLS & HOTEL ROANOKE 2	0	0	0	0	56	400795	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	56	400795	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	56	400795	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	56	400795	1
163	SALEM & 1 st	0	0	0	0	56	400795	1
164	SALEM & JEFFERSON	0	0	0	0	56	400795	1
165	JEFFERSON & CAMPBELL	0	0	0	0	56	400795	1
166	JEFFERSON & CHURCH	54570	54900	0	2	54	405806	1
167	JEFFERSON & FRANKLIN RD.	0	0	0	0	54	405806	1
168	JEFFERSON & BULLIT	0	0	0	0	54	405806	1
169	JEFFERSON & ELM	54840	0	8	0	62	406988	1
170	JEFFERSON & MOUNTAIN	0	0	0	0	62	406988	1
171	JEFFERSON & HIGHLAND	54915	0	0	1	61	407568	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	61	407568	1
173	JEFFERSON & MAPLE	0	0	0	0	61	407568	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	61	407568	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	61	407568	1
176	WALNUT & BELLEVIEW	0	0	0	0	61	407568	1
177	BELLEVIEW & LAUREL	0	0	0	0	61	407568	1
178	BELLEVIEW & PINK	0	0	0	0	61	407568	1
179	BELLEVIEW & LINDEN	0	0	0	0	61	407568	1
180	BELLEVIEW & ASH	0	0	0	0	61	407568	1
181	ROANOKE MEMORIAL HOSPITAL	0	55320	0	0	61	407568	1
182	JEFFERSON & WELLER	0	0	0	0	61	407568	1

183	JEFFERSON & MC CLANAHAN	0	0	0	0	61	407568	1
184	JEFFERSON & 22 nd	0	0	0	0	61	407568	1
185	JEFFERSON & 23 rd	0	0	0	0	61	407568	1
186	JEFFERSON & 24 th	0	0	0	0	61	407568	1
187	JEFFERSON & 25 th	0	0	0	0	61	407568	1
188	JEFFERSON & 26 th	0	0	0	0	61	407568	1
189	26 th & RICHELIEU	0	0	0	0	61	407568	1
190	26 th & CRYSTAL SPRING	0	0	0	0	61	407568	1
191	26 th & ROSALIND	0	0	0	0	61	407568	1
192	26 th & CAROLINA	0	0	0	0	61	407568	1
193	26 th & WYCLIFF	0	0	0	0	61	407568	1
194	26 th & AVENHAM	0	0	0	0	61	407568	1
195	AVENHAM & 27 th	0	0	0	0	61	407568	1
196	AVENHAM & 28 th	0	0	0	0	61	407568	1
197	AVENHAM & 29 th	0	0	0	0	61	407568	1
198	AVENHAM & WILDWOOD	0	0	0	0	61	407568	1
199	AVENHAM & CASSELL	0	0	0	0	61	407568	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	61	407568	1
201	AVENHAM &	0	0	0	0	61	407568	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	61	407568	1
203	AVENHAM & FRANKLIN ROAD	0	0	0	0	61	407568	1
204	FRANKLIN RD. & DUKE OF GL	0	0	0	0	61	407568	1
205	FRANKLIN RD. & K-MART	0	0	0	0	61	407568	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	61	407568	1
207	FRANKLIN RD. & PENARTH	55620	0	0	1	60	427076	1
208	SUPER X	0	0	0	0	60	427076	1
209	PENNEY'S	55860	0	2	1	61	430647	1
210		1	2			3540		
1	17 12 24 85	2						
1	BRENDLES (Time Point)	56550	56400	13	2	72	430658	1
2	OGDEN & FRANKLIN ROAD	0	0	0	0	72	430658	1
3	FRANKLIN RD. & ELM VIEW R	0	0	0	0	72	430658	1
4	FRANKLIN RD. & AVENHAM MA	0	0	0	0	72	430658	1
5	FRANKLIN RD. & TANGLEWOOD	0	0	0	0	72	430658	1
6	FRANKLIN RD. & PENARTH 1	56925	0	2	0	74	435564	1
7	FRANKLIN RD. & PENARTH 2	56955	0	1	0	75	435842	1
8	FRANKLIN RD. & TOWNSIDE	0	0	0	0	75	435842	1
9	FRANKLIN RD. & K-MART	0	0	0	0	75	435842	1
10	FRANKLIN RD. & DUKE OF GL	0	0	0	0	75	435842	1
11	FRANKLIN RD. & WILLOW OAK	0	0	0	0	75	435842	1
12	AVENHAM & DILLARD	0	0	0	0	75	435842	1
13	AVENHAM & WHITE OAK (2)	57060	0	1	0	76	438301	1
14	AVENHAM &	0	0	0	0	76	438301	1
15	AVENHAM & CLYDESDALE	0	0	0	0	76	438301	1
16	AVENHAM & CASSELL	0	0	0	0	76	438301	1
17	AVENHAM & AUDUBON	0	0	0	0	76	438301	1
18	AVENHAM & 29 th	0	0	0	0	76	438301	1
19	AVENHAM & 28 th	57180	0	1	0	77	441502	1
20	AVENHAM & 27 th	0	0	0	0	77	441502	1
21	AVENHAM & 26 th	0	0	0	0	77	441502	1
22	26 th & WYCLIFFE	0	0	0	0	77	441502	1

23	26 th & CAROLINA	0	0	0	0	77	441502	1
24	26 th & ROSALIND	0	0	0	0	77	441502	1
25	26 th & CRYSTAL SPRING	0	0	0	0	77	441502	1
26	26 th & RICHELIEU	0	0	0	0	77	441502	1
27	26 th & JEFFERSON	0	0	0	0	77	441502	1
28	JEFFERSON & 25 th	0	0	0	0	77	441502	1
29	JEFFERSON & 24 th	0	0	0	0	77	441502	1
30	JEFFERSON & 23 rd	0	0	0	0	77	441502	1
31	JEFFERSON & 22 nd	0	0	0	0	77	441502	1
32	JEFFERSON & YELLOW MT.	0	0	0	0	77	441502	1
33	JEFFERSON & WELLER	0	0	0	0	77	441502	1
34	ROANOKE MEMORIAL HOSPITAL	57405	57300	2	0	79	448043	1
35	BELLEVIEW & ASH	0	0	0	0	79	448043	1
36	BELLEVIEW & THYME	0	0	0	0	79	448043	1
37	BELLEVIEW & LINDEN	0	0	0	0	79	448043	1
38	BELLEVIEW & LAUREL	0	0	0	0	79	448043	1
39	BELLEVIEW & WALNUT	0	0	0	0	79	448043	1
40	WALNUT AVENUE BRIDGE 2	0	0	0	0	79	448043	1
41	WALNUT AVENUE BRIDGE 1	0	0	0	0	79	448043	1
42	JEFFERSON & MAPLE	0	0	0	0	79	448043	1
43	JEFFERSON & ALBEMARLE	57615	0	1	0	80	453378	1
44	JEFFERSON & HIGHLAND	0	0	0	0	80	453378	1
45	JEFFERSON & MOUNTAIN	0	0	0	0	80	453378	1
46	JEFFERSON & ELM	57720	0	0	1	79	454653	1
47	JEFFERSON & BULLIT	0	0	0	0	79	454653	1
48	JEFFERSON & FRANKLIN RD.	0	0	0	0	79	454653	1
49	JEFFERSON & CHURCH	57810	57900	2	7	74	455975	1
50	JEFFERSON & CAMPBELL	57885	0	3	0	77	456369	1
51	CAMPBELL & MARKET ST.	58005	0	1	1	77	456995	1
52	WILLIAMSON ROAD & SHENAND	0	0	0	0	77	456995	1
53	WELLS & COMMONWEALTH	0	0	0	0	77	456995	1
54	WELLS & HOTEL ROANOKE 1	0	0	0	0	77	456995	1
55	WELLS & JEFFERSON	0	0	0	0	77	456995	1
56	JEFFERSON & GILMER	0	0	0	0	77	456995	1
57	GILMER & GAINSBORO	0	0	0	0	77	456995	1
58	GAINSBORO & PATTON	0	0	0	0	77	456995	1
59	GAINSBORO & HARRISON	0	0	0	0	77	456995	1
60	GAINSBORO & MADISON	0	0	0	0	77	456995	1
61	GAINSBORO & McDOWELL	0	0	0	0	77	456995	1
62	GAINSBORO & ORANGE	0	0	0	0	77	456995	1
63	ORANGE & QUARRY	0	0	0	0	77	456995	1
64	ORANGE & 5 th	0	0	0	0	77	456995	1
65	ORANGE & LUCY ADDISON JR.	0	0	0	0	77	456995	1
66	ORANGE & 8 th	0	0	0	0	77	456995	1
67	ORANGE & 10 th	0	58380	0	0	77	456995	1
68	ORANGE & 11 th	58410	0	1	0	78	466644	1
69	11 th & HANNOVER	0	0	0	0	78	466644	1
70	11 th & STAUNTON	0	0	0	0	78	466644	1
71	STAUNTON & 12 th	58485	0	0	1	77	467595	1
72	STAUNTON & 13 th	0	0	0	0	77	467595	1
73	STAUNTON & 14 th	58515	0	0	1	76	468314	1

74	STAUNTON & EUREKA CIRCLE	0	0	0	0	76	468314	1
75	STAUNTON & 15 th	0	0	0	0	76	468314	1
76	STAUNTON & 16 th	58575	0	0	1	75	469404	1
77	STAUNTON & 17 th	0	0	0	0	75	469404	1
78	STAUNTON & 18 th	0	0	0	0	75	469404	1
79	STAUNTON & 19 th	0	0	0	0	75	469404	1
80	STAUNTON & 20 th	0	0	0	0	75	469404	1
81	STAUNTON & 21 st	0	0	0	0	75	469404	1
82	STAUNTON & 22 nd	0	0	0	0	75	469404	1
83	STAUNTON & 23 rd	0	0	0	0	75	469404	1
84	STAUNTON & 24 th	58725	0	0	1	74	473092	1
85	STAUNTON & LAFAYETTE	0	0	0	0	74	473092	1
86	LAFAYETTE & NEW YORK	58800	0	0	1	73	474112	1
87	LAFAYETTE & CLIFTON	0	0	0	0	73	474112	1
88	LAFAYETTE & DELAWARE	58860	0	1	2	72	474715	1
89	LAFAYETTE & MASSACHUSSETS	0	0	0	0	72	474715	1
90	LAFAYETTE & FLORIDA	0	0	0	0	72	474715	1
91	LAFAYETTE &	0	0	0	0	72	474715	1
92	LAFAYETTE & COVE ROAD	0	0	0	0	72	474715	1
93	COVE ROAD & FAIRLAND	0	0	0	0	72	474715	1
94	COVE ROAD & ASPEN	58965	0	0	1	71	477893	1
95	COVE ROAD & WELLSLEY	0	0	0	0	71	477893	1
96	COVE ROAD & ABBOTT	0	0	0	0	71	477893	1
97	COVE ROAD & COVELAND	0	0	0	0	71	477893	1
98	COVE ROAD & GOLFSIDE	0	0	0	0	71	477893	1
99	COVE ROAD & GUILDHALL	0	0	0	0	71	477893	1
100	COVE ROAD & HERSHBERGER	0	0	0	0	71	477893	1
101	COVE ROAD &	0	0	0	0	71	477893	1
102	COVE ROAD &	0	0	0	0	71	477893	1
103	COVE ROAD & ROUTT	0	0	0	0	71	477893	1
104	ROUTT & BLACK OAK	0	0	0	0	71	477893	1
105	ROUTT & FERNCLIFF	0	0	0	0	71	477893	1
106	FERNCLIFF APARTMENTS N.	0	59100	0	0	71	477893	1
107	FERNCLIFF APARTMENTS S.	0	0	0	0	71	477893	1
108	FERNCLIFF &	0	0	0	0	71	477893	1
109	FERNCLIFF &	59325	0	2	1	72	486754	1
110	FERNCLIFF & HERSHBERGER	0	0	0	0	72	486754	1
111	HERSHBERGER & SWARTHMORE	0	0	0	0	72	486754	1
112	HERSHBERGER & COVE ROAD	0	0	0	0	72	486754	1
113	COVE ROAD & GUILDHALL	0	0	0	0	72	486754	1
114	COVE ROAD & GOLFSIDE	0	0	0	0	72	486754	1
115	COVE ROAD & FRESNO	0	0	0	0	72	486754	1
116	COVE ROAD & ABBOTT	0	0	0	0	72	486754	1
117	COVE ROAD & WELLSLEY	0	0	0	0	72	486754	1
118	COVE ROAD & ASPEN	0	0	0	0	72	486754	1
119	COVE ROAD & FAIRLAND	0	0	0	0	72	486754	1
120	COVE ROAD & LAFAYETTE	0	0	0	0	72	486754	1
121	LAFAYETTE &	0	0	0	0	72	486754	1
122	LAFAYETTE & FLORIDA	0	0	0	0	72	486754	1
123	LAFAYETTE & MASSACHUSSETS	59565	0	0	1	71	494362	1
124	LAFAYETTE & DELAWARE	0	0	0	0	71	494362	1

125	LAFAYETTE & CLIFTON	0	0	0	0	71	494362	1
126	LAFAYETTE & NEW YORK	0	0	0	0	71	494362	1
127	LAFAYETTE & STAUNTON	0	0	0	0	71	494362	1
128	STAUNTON & 24 th	0	0	0	0	71	494362	1
129	STAUNTON & 23 rd	59670	0	1	0	72	496960	1
130	STAUNTON & 22 nd	0	0	0	0	72	496960	1
131	STAUNTON & 21 st	0	0	0	0	72	496960	1
132	STAUNTON & 20 th	0	0	0	0	72	496960	1
133	STAUNTON & 19 th	0	0	0	0	72	496960	1
134	STAUNTON & 18 th	0	0	0	0	72	496960	1
135	STAUNTON & 17 th	0	0	0	0	72	496960	1
136	STAUNTON & 16 th	59805	0	0	1	71	500115	1
137	STAUNTON & 15 th	0	0	0	0	71	500115	1
138	STAUNTON & EUREKA CIRCLE	0	0	0	0	71	500115	1
139	STAUNTON & 14 th	0	0	0	0	71	500115	1
140	STAUNTON & 13 th	0	0	0	0	71	500115	1
141	STAUNTON & 12 th	0	0	0	0	71	500115	1
142	STAUNTON & 11 th	0	0	0	0	71	500115	1
143	11 th & HANOVER	0	0	0	0	71	500115	1
144	ORANGE & 10 th	0	59820	0	0	71	500115	1
145	ORANGE &	0	0	0	0	71	500115	1
146	ORANGE & 8 th	0	0	0	0	71	500115	1
147	ORANGE & LUCY ADDISON JR.	0	0	0	0	71	500115	1
148	ORANGE & 5 th	0	0	0	0	71	500115	1
149	ORANGE & QUARRY	0	0	0	0	71	500115	1
150	ORANGE & GAINSBORO	60015	0	1	0	72	507329	1
151	GAINSBORO & McDOWELL	0	0	0	0	72	507329	1
152	GAINSBORO & MADISON	0	0	0	0	72	507329	1
153	GAINSBORO & HARRISON	0	0	0	0	72	507329	1
154	GAINSBORO & PATTON	60105	0	2	0	74	509138	1
155	1 st & GILMER	0	0	0	0	74	509138	1
156	GILMER & JEFFERSON	0	0	0	0	74	509138	1
157	JEFFERSON & WELLS	0	0	0	0	74	509138	1
158	WELLS & HOTEL ROANOKE 1	0	0	0	0	74	509138	1
159	WELLS & HOTEL ROANOKE 2	0	0	0	0	74	509138	1
160	WELLS & WILLIAMSON ROAD	0	0	0	0	74	509138	1
161	WILLIAMSON ROAD & SHENAND	0	0	0	0	74	509138	1
162	WILLIAMSON ROAD & SALEM	0	0	0	0	74	509138	1
163	SALEM & 1 st	0	0	0	0	74	509138	1
164	SALEM & JEFFERSON	60330	0	0	2	72	513058	1
165	JEFFERSON & CAMPBELL	60540	0	3	1	74	513498	1
166	JEFFERSON & CHURCH	0	60600	0	0	74	513498	1
167	JEFFERSON & FRANKLIN RD.	0	0	0	0	74	513498	1
168	JEFFERSON & BULLIT	0	0	0	0	74	513498	1
169	JEFFERSON & ELM	0	0	0	0	74	513498	1
170	JEFFERSON & MOUNTAIN	0	0	0	0	74	513498	1
171	JEFFERSON & HIGHLAND	0	0	0	0	74	513498	1
172	JEFFERSON & ALBEMARLE	0	0	0	0	74	513498	1
173	JEFFERSON & MAPLE	0	0	0	0	74	513498	1
174	WALNUT AVENUE BRIDGE 1	0	0	0	0	74	513498	1
175	WALNUT AVENUE BRIDGE 2	0	0	0	0	74	513498	1

176	WALNUT & BELLEVIEW	0	0	0	0	74	513498	1
177	BELLEVIEW & LAUREL	0	0	0	0	74	513498	1
178	BELLEVIEW & PINK	0	0	0	0	74	513498	1
179	BELLEVIEW & LINDEN	0	0	0	0	74	513498	1
180	BELLEVIEW & ASH	0	0	0	0	74	513498	1
181	ROANOKE MEMORIAL HOSPITAL	0	61020	0	0	74	513498	1
182	JEFFERSON & WELLER	0	0	0	0	74	513498	1
183	JEFFERSON & MC CLANAHAN	0	0	0	0	74	513498	1
184	JEFFERSON & 22 nd	0	0	0	0	74	513498	1
185	JEFFERSON & 23 rd	0	0	0	0	74	513498	1
186	JEFFERSON & 24 th	0	0	0	0	74	513498	1
187	JEFFERSON & 25 th	0	0	0	0	74	513498	1
188	JEFFERSON & 26 th	0	0	0	0	74	513498	1
189	26 th & RICHELIEU	0	0	0	0	74	513498	1
190	26 th & CRYSTAL SPRING	0	0	0	0	74	513498	1
191	26 th & ROSALIND	0	0	0	0	74	513498	1
192	26 th & CAROLINA	0	0	0	0	74	513498	1
193	26 th & WYCLIFF	0	0	0	0	74	513498	1
194	26 th & AVENHAM	0	0	0	0	74	513498	1
195	AVENHAM & 27 th	0	0	0	0	74	513498	1
196	AVENHAM & 28 th	0	0	0	0	74	513498	1
197	AVENHAM & 29 th	0	0	0	0	74	513498	1
198	AVENHAM & WILDWOOD	0	0	0	0	74	513498	1
199	AVENHAM & CASSELL	0	0	0	0	74	513498	1
200	AVENHAM & WHITE OAK (1)	0	0	0	0	74	513498	1
201	AVENHAM &	0	0	0	0	74	513498	1
202	AVENHAM & WHITE OAK (2)	0	0	0	0	74	513498	1
203	AVENHAM & FRANKLIN ROAD	61200	0	0	1	73	532334	1
204	FRANKLIN RD. & DUKE OF GL	0	0	0	0	73	532334	1
205	FRANKLIN RD. & K-MART	0	0	0	0	73	532334	1
206	FRANKLIN RD. & TOWNSIDE	0	0	0	0	73	532334	1
207	FRANKLIN RD. & PENARTH	0	0	0	0	73	532334	1
208	SUPER X	0	0	0	0	73	532334	1
209	PENNEY'S	0	0	0	0	73	532334	1
210		1	2		3900			

ROANOKE VALLEY-METRO APC SYSTEM

ROUTE	BLOCK	NUMBER OF TIMES AROUND	DATE	BUS #	ACCUMULATED DISTANCE	ACCUMULATED IDLE	LOAD	DAY
1	17	1	12/19/85	1	.08	3: 0	0	1

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	13:31:30	13:30: 0	5	2	3	.08		180	NO
3	FRANKLIN RD. & ELM VIEW R	13:34:45	0: 0: 0	0	1	2	.40		60	NO
7	FRANKLIN RD. & PENARTH 2	13:38: 0	0: 0: 0	1	1	2	1.04		0	NO
10	FRANKLIN RD. & DUKE OF GL	13:38:45	0: 0: 0	0	1	1	1.24	*	0	NO
18	AVENHAM & 29 th	13:40:45	0: 0: 0	1	0	2	2.02		0	NO
21	AVENHAM & 26 th	13:41:30	0: 0: 0	1	0	3	2.27		0	NO
23	26 th & CAROLINA	13:42:30	0: 0: 0	1	0	4	2.42		0	NO
24	26 th & ROSALIND	13:43: 0	0: 0: 0	2	0	6	2.49		0	NO
28	JEFFERSON & 25 th	13:44:15	0: 0: 0	1	0	7	2.78	*	0	NO
32	JEFFERSON & YELLOW MT.	13:45:30	0: 0: 0	3	0	10	3.10		0	NO
43	JEFFERSON & ALBEMARLE	13:49:45	0: 0: 0	1	0	11	4.34		0	NO
47	JEFFERSON & BULLIT	13:51:30	0: 0: 0	0	1	10	4.69	*	0	NO
49	JEFFERSON & CHURCH	13:56:30	13:55: 0	3	4	9	4.83		270	NO
50	JEFFERSON & CAMPBELL	13:57:30	0: 0: 0	3	0	12	4.91	*	0	NO
51	CAMPBELL & MARKET ST.	13:58:45	0: 0: 0	3	0	15	5.03		0	NO
57	GILMER & GAINSBORO	14: 3:45	0: 0: 0	0	1	14	5.60		60	NO
71	STAUNTON & 12 th	14: 8: 0	0: 0: 0	0	1	13	7.04		0	NO
78	STAUNTON & 18 th	14: 9:30	0: 0: 0	1	0	14	7.55		0	NO
86	LAFAYETTE & NEW YORK	14:11:45	0: 0: 0	0	1	13	8.28	*	0	NO
106	FERNCLIFF APARTMENTS N.	0: 0: 0	14:15: 0	0	0	13	8.28		0	NO
107	FERNCLIFF APARTMENTS S.	14:18: 0	0: 0: 0	1	0	14	10.48		60	NO
108	FERNCLIFF &	14:18:30	0: 0: 0	0	1	13	10.59	*	0	NO
109	FERNCLIFF &	14:19:15	0: 0: 0	3	0	16	10.66		0	NO
111	HERSHBERGER & SWARTHMORE	14:20:15	0: 0: 0	1	0	17	10.87		0	NO
125	LAFAYETTE & CLIFTON	14:24:15	0: 0: 0	0	1	16	12.25		0	NO
126	LAFAYETTE & NEW YORK	14:24:45	0: 0: 0	1	0	17	12.32	*	0	NO
134	STAUNTON & 18 th	14:27: 0	0: 0: 0	0	1	16	13.02		0	NO
143	11 th & HANOVER	14:30: 0	0: 0: 0	2	1	17	13.70	*	60	NO
164	SALEM & JEFFERSON	14:37:15	0: 0: 0	1	0	18	15.62		0	NO
166	JEFFERSON & CHURCH	14:41: 0	14:40: 0	5	1	22	15.73		60	NO
170	JEFFERSON & MOUNTAIN	14:42:15	0: 0: 0	0	1	21	15.99		0	NO
186	JEFFERSON & 24 th	14:48:15	0: 0: 0	0	1	20	17.65		0	NO
187	JEFFERSON & 25 th	14:49: 0	0: 0: 0	1	0	21	17.73		0	NO
189	26 th & RICHELIEU	14:49:30	0: 0: 0	0	1	20	17.90		0	NO
209	PENNEY'S	14:57:30	0: 0: 0	2	0	22	20.29		0	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	15: 3:15	15: 0: 0	6	5	23	20.29		750	NO
2	OGDEN & FRANKLIN ROAD	15: 6:30	0: 0: 0	1	0	24	20.59		60	NO
7	FRANKLIN RD. & PENARTH 2	15: 8:45	0: 0: 0	1	0	25	21.29		0	NO
9	FRANKLIN RD. & K-MART	15: 9:15	0: 0: 0	1	0	26	21.41		0	NO
13	AVENHAM & WHITE OAK (2)	15:10:45	0: 0: 0	5	0	31	21.74	*	0	NO
15	AVENHAM & CLYDESDALE	15:11:45	0: 0: 0	1	0	32	21.98		0	NO

16	AVENHAM & CASSELL	15:12:15	0: 0: 0	1	0	33	22.06		0	NO
18	AVENHAM & 29 th	15:13: 0	0: 0: 0	1	0	34	22.25		0	NO
23	26 th & CAROLINA	15:14:15	0: 0: 0	1	0	35	22.65	*	0	NO
32	JEFFERSON & YELLOW MT.	15:16:30	0: 0: 0	0	1	34	23.34		0	NO
34	ROANOKE MEMORIAL HOSPITAL	15:18: 0	15:15: 0	2	0	36	23.56		0	NO
49	JEFFERSON & CHURCH	15:25: 0	15:25: 0	4	4	36	25.06		60	NO
50	JEFFERSON & CAMPBELL	15:26:15	0: 0: 0	2	1	37	25.15	*	0	NO
51	CAMPBELL & MARKET ST.	15:28: 0	0: 0: 0	6	0	43	25.27		0	NO
55	WELLS & JEFFERSON	15:30:30	0: 0: 0	1	0	44	25.74		0	NO
70	11 th & STAUNTON	15:36:15	0: 0: 0	0	1	43	27.18		0	NO
71	STAUNTON & 12 th	15:36:45	0: 0: 0	0	1	42	27.26	*	0	NO
84	STAUNTON & 24 th	15:40:45	0: 0: 0	1	0	43	28.30		0	NO
97	COVE ROAD & COVELAND	15:44: 0	0: 0: 0	1	1	43	29.44		0	NO
111	HERSHBERGER & SWARTHMORE	15:48: 0	0: 0: 0	0	1	42	31.10		60	NO
114	COVE ROAD & GOLFSIDE	15:49:15	0: 0: 0	3	0	45	31.43		0	NO
130	STAUNTON & 22 nd	15:53: 0	0: 0: 0	1	0	46	32.88		0	NO
138	STAUNTON & EUREKA CIRCLE	15:55:15	0: 0: 0	1	0	47	33.54		0	NO
142	STAUNTON & 11 th	15:56:30	0: 0: 0	1	0	48	33.80		0	NO
148	ORANGE & 5 th	15:58:15	0: 0: 0	1	0	49	34.48		0	NO
159	WELLS & HOTEL ROANOKE 2	16: 2:15	0: 0: 0	1	0	50	35.48		0	NO
164	SALEM & JEFFERSON	16: 3:30	0: 0: 0	0	1	49	35.84		0	NO
165	JEFFERSON & CAMPBELL	16: 4:15	0: 0: 0	1	0	50	35.94		0	NO
184	JEFFERSON & 22 nd	16:11: 0	0: 0: 0	1	0	51	37.79		60	NO

ROANOKE VALLEY-METRO APC SYSTEM

ROUTE	BLOCK	NUMBER OF TIMES AROUND	DATE	BUS #	ACCUMULATED DISTANCE	IDLE	LOAD	DAY
1	17	1	12/24/85	1	.57	2:45	0	2

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	9:27:45	9:25: 0	1	0	1	.78		225	NO
2	OGDEN & FRANKLIN ROAD	9:28:45	0: 0: 0	1	0	2	1.01		0	NO
21	AVENHAM & 26 th	9:34:15	0: 0: 0	1	0	3	2.96		0	NO
32	JEFFERSON & YELLOW MT.	9:36:45	0: 0: 0	1	0	4	3.87		0	NO
33	JEFFERSON & WELLER	9:37:15	0: 0: 0	1	0	5	3.96		0	NO
46	JEFFERSON & ELM	9:42:15	0: 0: 0	0	1	4	5.31		0	NO
48	JEFFERSON & FRANKLIN RD.	9:49:30	0: 0: 0	0	1	3	5.45		270	NO
49	JEFFERSON & CHURCH	9:50:30	9:50: 0	1	0	4	5.54	*	0	NO
50	JEFFERSON & CAMPBELL	9:51:45	0: 0: 0	4	0	8	5.66		0	NO
62	GAINSBORO & ORANGE	9:57:30	0: 0: 0	1	0	9	6.79		0	NO
67	ORANGE & 10 th	9:58:30	9:58: 0	2	0	11	7.47		0	NO
85	STAUNTON & LAFAYETTE	10: 3: 0	0: 0: 0	1	0	12	8.90		0	NO
105	ROUTT & FERNCLIFF	10: 9:15	0: 0: 0	1	1	12	10.95		60	NO
108	FERNCLIFF &	10:10:45	0: 0: 0	0	5	7	11.30		0	NO
112	HERSHBERGER & COVE ROAD	10:12:45	0: 0: 0	1	0	8	11.79		0	NO
121	LAFAYETTE &	10:15:30	0: 0: 0	1	0	9	12.58		0	NO
123	LAFAYETTE & MASSACHUSSETS	10:16:15	0: 0: 0	2	0	11	12.74		0	NO
124	LAFAYETTE & DELAWARE	10:17: 0	0: 0: 0	2	0	13	12.82		0	NO
126	LAFAYETTE & NEW YORK	10:18:15	0: 0: 0	2	0	15	12.94		0	NO
128	STAUNTON & 24 th	10:19: 0	0: 0: 0	2	0	17	13.14		0	NO
129	STAUNTON & 23 rd	10:19:30	0: 0: 0	2	0	19	13.22	*	0	NO
143	11 th & HANOVER	10:22:45	0: 0: 0	2	0	21	14.33	*	0	NO
156	GILMER & JEFFERSON	10:27:30	0: 0: 0	1	0	22	15.64		60	NO
163	SALEM & 1 st	10:29:15	0: 0: 0	0	1	21	16.14		0	NO
164	SALEM & JEFFERSON	10:30:45	0: 0: 0	1	4	18	16.27	*	0	NO
165	JEFFERSON & CAMPBELL	10:34:30	0: 0: 0	3	0	21	16.36		210	NO
170	JEFFERSON & MOUNTAIN	10:36:45	0: 0: 0	1	0	22	16.68		0	NO
204	FRANKLIN RD. & DUKE OF GL	10:47: 0	0: 0: 0	1	0	23	20.01		60	NO
208	SUPER X	10:50:15	0: 0: 0	0	2	21	20.85		0	NO
209	PENNEY'S	10:51:45	0: 0: 0	1	6	16	20.93		0	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	10:59:15	10:55: 0	1	5	12	20.94		1200	NO
32	JEFFERSON & YELLOW MT.	11: 9: 0	0: 0: 0	2	0	14	23.98		0	NO
34	ROANOKE MEMORIAL HOSPITAL	11: 9:45	11:10: 0	1	0	15	24.21		0	NO
49	JEFFERSON & CHURCH	11:22: 0	11:20: 0	2	4	13	25.69		420	NO
50	JEFFERSON & CAMPBELL	11:23:15	0: 0: 0	3	0	16	25.78	*	0	NO
51	CAMPBELL & MARKET ST.	11:25: 0	0: 0: 0	3	0	19	25.89	*	0	NO
58	GAINSBORO & PATTON	11:27:45	0: 0: 0	1	0	20	26.54		0	NO
70	11 th & STAUNTON	11:32:30	0: 0: 0	0	1	19	27.81		60	NO
78	STAUNTON & 18 th	11:33:45	0: 0: 0	0	1	18	28.40		0	NO
106	FERNCLIFF APARTMENTS N.	11:44:45	11:40: 0	1	1	18	31.18		285	NO
109	FERNCLIFF &	11:46:30	0: 0: 0	5	2	21	31.53	*	0	NO
112	HERSHBERGER & COVE ROAD	11:48: 0	0: 0: 0	2	0	23	31.92		0	NO
123	LAFAYETTE & MASSACHUSSETS	11:50:15	0: 0: 0	0	1	22	32.97	*	0	NO
129	STAUNTON & 23 rd	11:51:45	0: 0: 0	0	1	21	33.46		0	NO
146	ORANGE & 8 th	11:56:30	0: 0: 0	1	0	22	34.87		60	NO
154	GAINSBORO & PATTON	11:58:45	0: 0: 0	0	2	20	35.77	*	0	NO

157	JEFFERSON & WELLS	11:59:30	0: 0: 0	0	1	19	35.93		0	NO
163	SALEM & 1 st	12: 2:30	0: 0: 0	0	1	18	36.36		60	NO
165	JEFFERSON & CAMPBELL	12: 6:45	0: 0: 0	11	0	29	36.58		60	NO
166	JEFFERSON & CHURCH	12: 8:30	12:10: 0	1	0	30	36.59		60	NO
170	JEFFERSON & MOUNTAIN	12:10:15	0: 0: 0	3	0	33	36.91		0	NO
183	JEFFERSON & MC CLANAHAN	12:14:30	0: 0: 0	1	0	34	38.33	*	0	NO
184	JEFFERSON & 22 nd	12:15: 0	0: 0: 0	0	1	33	38.43	*	0	NO
186	JEFFERSON & 24 th	12:15:45	0: 0: 0	1	0	34	38.60		0	NO
207	FRANKLIN RD. & PENARTH	12:22:30	0: 0: 0	0	1	33	40.53		0	NO
208	SUPER X	12:24:45	0: 0: 0	2	1	34	41.06		0	NO
209	PENNEY'S	12:26: 0	0: 0: 0	0	6	28	41.15		0	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	12:31: 0	12:30: 0	6	1	33	41.16		2415	NO
13	AVENHAM & WHITE OAK (2)	12:36:45	0: 0: 0	0	1	32	42.62		0	NO
15	AVENHAM & CLYDESDALE	12:37:30	0: 0: 0	1	0	33	42.87	*	0	NO
17	AVENHAM & AUDUBON	12:38:15	0: 0: 0	1	0	34	43.03		0	NO
22	26 th & WYCLIFFE	12:39:30	0: 0: 0	1	0	35	43.47	*	0	NO
32	JEFFERSON & YELLOW MT.	12:41:30	0: 0: 0	1	0	36	44.23		0	NO
34	ROANOKE MEMORIAL HOSPITAL	12:45: 0	12:45: 0	2	0	38	44.46	*	60	NO
42	JEFFERSON & MAPLE	12:49: 0	0: 0: 0	0	1	37	45.40		0	NO
49	JEFFERSON & CHURCH	12:53:45	12:55: 0	3	5	35	45.95		135	NO
51	CAMPBELL & MARKET ST.	12:57:15	0: 0: 0	7	0	42	46.15		60	NO
67	ORANGE & 10 th	13: 3:15	13: 3: 0	1	0	43	47.84		0	NO
68	ORANGE & 11 th	13: 3:45	0: 0: 0	0	1	42	47.97		0	NO
72	STAUNTON & 13 th	13: 5: 0	0: 0: 0	1	0	43	48.23		0	NO
73	STAUNTON & 14 th	13: 5:30	0: 0: 0	0	1	42	48.29	*	0	NO
83	STAUNTON & 23 rd	13: 7:45	0: 0: 0	0	2	40	49.09		0	NO
88	LAFAYETTE & DELAWARE	13: 9:30	0: 0: 0	0	2	38	49.50		0	NO
97	COVE ROAD & COVELAND	13:11:45	0: 0: 0	0	1	37	50.33		0	NO
106	FERNCLIFF APARTMENTS N.	13:15:30	13:15: 0	1	1	37	51.44		0	NO
108	FERNCLIFF &	13:16:15	0: 0: 0	0	1	36	51.70	*	0	NO
113	COVE ROAD & GUILDHALL	13:18:15	0: 0: 0	1	0	37	52.28		0	NO
117	COVE ROAD & WELLSLEY	13:19:15	0: 0: 0	1	0	38	52.62		0	NO
122	LAFAYETTE & FLORIDA	13:20:30	0: 0: 0	1	0	39	53.07	*	0	NO
123	LAFAYETTE & MASSACHUSSETS	13:21: 0	0: 0: 0	1	0	40	53.21		0	NO
129	STAUNTON & 23 rd	13:22:45	0: 0: 0	1	0	41	53.71		0	NO
132	STAUNTON & 20 th	13:23:30	0: 0: 0	0	1	40	53.96		0	NO
143	11 th & HANOVER	13:26: 0	0: 0: 0	3	0	43	54.81	*	0	NO
154	GAINSBORO & PATTON	13:29:45	0: 0: 0	0	1	42	56.01		0	NO
163	SALEM & 1 st	13:32:30	0: 0: 0	1	0	43	56.62	*	0	NO
164	SALEM & JEFFERSON	13:33:30	0: 0: 0	0	1	42	56.74		0	NO
165	JEFFERSON & CAMPBELL	13:35:15	0: 0: 0	6	0	48	56.83		60	NO
166	JEFFERSON & CHURCH	13:39:15	13:40: 0	4	2	50	56.83		240	NO
167	JEFFERSON & FRANKLIN RD.	13:39:45	0: 0: 0	2	0	52	56.86		0	NO
169	JEFFERSON & ELM	13:41: 0	0: 0: 0	0	1	51	57.04		0	NO
181	ROANOKE MEMORIAL HOSPITAL	13:45:30	13:47: 0	0	1	50	58.32		0	NO
184	JEFFERSON & 22 nd	13:47: 0	0: 0: 0	1	1	50	58.67		0	NO
194	26 th & AVENHAM	13:50: 0	0: 0: 0	0	1	49	59.43		0	NO
204	FRANKLIN RD. & DUKE OF GL	13:53:15	0: 0: 0	0	2	47	60.47		0	NO
208	SUPER X	13:56:30	0: 0: 0	0	2	45	61.30		0	NO
209	PENNEY'S	13:58:30	0: 0: 0	6	4	47	61.39		60	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	14: 1:15	14: 0: 0	3	0	50	61.39		3090	NO
5	FRANKLIN RD. & TANGLEWOOD	14: 4:45	0: 0: 0	0	1	49	61.93		0	NO
7	FRANKLIN RD. & PENARTH 2	14: 7:15	0: 0: 0	1	0	50	62.39		0	NO
10	FRANKLIN RD. & DUKE OF GL	14: 8: 0	0: 0: 0	0	1	49	62.58		0	NO

16	AVENHAM & CASSELL	14: 9:45	0:-0: 0	1	0	50	63.18		0	NO
31	JEFFERSON & 22 nd	14:13:15	0: 0: 0	0	1	49	64.40		0	NO
32	JEFFERSON & YELLOW MT.	14:14:15	0: 0: 0	2	0	51	64.46		0	NO
33	JEFFERSON & WELLER	14:15: 0	0: 0: 0	1	0	52	64.60		0	NO
45	JEFFERSON & MOUNTAIN	14:19:15	0: 0: 0	0	3	49	65.87		0	NO
49	JEFFERSON & CHURCH	14:22:45	14:25: 0	2	3	48	66.18		0	NO
50	JEFFERSON & CAMPBELL	14:23:45	0: 0: 0	3	0	51	66.26		0	NO
51	CAMPBELL & MARKET ST.	14:24:30	0: 0: 0	0	1	50	66.27		0	NO
52	WILLIAMSON ROAD & SHENAND	14:25:15	0: 0: 0	3	0	53	66.37		0	NO
59	GAINSBORO & HARRISON	14:28:15	0: 0: 0	1	0	54	66.85		0	NO
67	ORANGE & 10 th	14:31:15	14:33: 0	1	0	55	67.91		0	NO
89	LAFAYETTE & MASSACHUSSETS	14:38: 0	0: 0: 0	0	2	53	69.62		0	NO
90	LAFAYETTE & FLORIDA	14:38:45	0: 0: 0	1	1	53	69.74		0	NO
97	COVE ROAD & COVELAND	14:40:45	0: 0: 0	0	1	52	70.34	*	0	NO
101	COVE ROAD &	14:42:15	0: 0: 0	0	1	51	70.86		0	NO
110	FERNCLIFF & HERSHBERGER	14:49:45	0: 0: 0	2	0	53	71.84		225	NO
112	HERSHBERGER & COVE ROAD	14:50:45	0: 0: 0	3	0	56	72.02		0	NO
114	COVE ROAD & GOLFSIDE	14:52: 0	0: 0: 0	0	1	55	72.22		0	NO
132	STAUNTON & 20 th	14:58: 0	0: 0: 0	1	0	56	73.87		0	NO
146	ORANGE & 8 th	15: 1:30	0: 0: 0	1	0	57	75.05		0	NO
148	ORANGE & 5 th	15: 4: 0	0: 0: 0	0	2	55	75.35		0	NO
153	GAINSBORO & HARRISON	15: 5:45	0: 0: 0	1	0	56	75.91		0	NO
166	JEFFERSON & CHURCH	15: 9:30	15:15: 0	0	2	54	76.86		0	NO
169	JEFFERSON & ELM	15:14: 0	0: 0: 0	8	0	62	77.08		165	NO
171	JEFFERSON & HIGHLAND	15:15:15	0: 0: 0	0	1	61	77.19		0	NO
207	FRANKLIN RD. & PENARTH	15:27: 0	0: 0: 0	0	1	60	80.89		0	NO
209	PENNEY'S	15:31: 0	0: 0: 0	2	1	61	81.56		60	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	BRENDLES (Time Point)	15:42:30	15:40: 0	13	2	72	81.56		3585	NO
6	FRANKLIN RD. & PENARTH 1	15:48:45	0: 0: 0	2	0	74	82.49		0	NO
7	FRANKLIN RD. & PENARTH 2	15:49:15	0: 0: 0	1	0	75	82.55		0	NO
13	AVENHAM & WHITE OAK (2)	15:51: 0	0: 0: 0	1	0	76	83.01	*	0	NO
19	AVENHAM & 28 th	15:53: 0	0: 0: 0	1	0	77	83.62	*	0	NO
34	ROANOKE MEMORIAL HOSPITAL	15:56:45	15:55: 0	2	0	79	84.86		0	NO
43	JEFFERSON & ALBEMARLE	16: 0:15	0: 0: 0	1	0	80	85.87		0	NO
46	JEFFERSON & ELM	16: 2: 0	0: 0: 0	0	1	79	86.11	*	60	NO
49	JEFFERSON & CHURCH	16: 3:30	16: 5: 0	2	7	74	86.36		60	NO
50	JEFFERSON & CAMPBELL	16: 4:45	0: 0: 0	3	0	77	86.43		0	NO
51	CAMPBELL & MARKET ST.	16: 6:45	0: 0: 0	1	1	77	86.55		0	NO
67	ORANGE & 10 th	0: 0: 0	16:13: 0	0	0	77	86.55		0	NO
68	ORANGE & 11 th	16:13:30	0: 0: 0	1	0	78	88.38		0	NO
71	STAUNTON & 12 th	16:14:45	0: 0: 0	0	1	77	88.56	*	0	NO
73	STAUNTON & 14 th	16:15:15	0: 0: 0	0	1	76	88.70		0	NO
76	STAUNTON & 16 th	16:16:15	0: 0: 0	0	1	75	88.90		0	NO
84	STAUNTON & 24 th	16:18:45	0: 0: 0	0	1	74	89.60		0	NO
86	LAFAYETTE & NEW YORK	16:20: 0	0: 0: 0	0	1	73	89.79		0	NO
88	LAFAYETTE & DELAWARE	16:21: 0	0: 0: 0	1	2	72	89.91	*	0	NO
94	COVE ROAD & ASPEN	16:22:45	0: 0: 0	0	1	71	90.51		0	NO
109	FERNCLIFF &	16:28:45	0: 0: 0	2	1	72	92.19		0	NO
123	LAFAYETTE & MASSACHUSSETS	16:32:45	0: 0: 0	0	1	71	93.63		0	NO
129	STAUNTON & 23 rd	16:34:30	0: 0: 0	1	0	72	94.12		0	NO
136	STAUNTON & 16 th	16:36:45	0: 0: 0	0	1	71	94.72		0	NO
150	ORANGE & GAINSBORO	16:40:15	0: 0: 0	1	0	72	96.09		0	NO
154	GAINSBORO & PATTON	16:41:45	0: 0: 0	2	0	74	96.43		0	NO
164	SALEM & JEFFERSON	16:45:30	0: 0: 0	0	2	72	97.17		0	NO
165	JEFFERSON & CAMPBELL	16:49: 0	0: 0: 0	3	1	74	97.25		195	NO
203	AVENHAM & FRANKLIN ROAD	17: 0: 0	0: 0: 0	0	1	73	100.82		0	NO

ROANOKE VALLEY-METRO APC SYSTEM

ROUTE	BLOCK	NUMBER OF TIMES AROUND	DATE	BUS #	ACCUMULATED DISTANCE	IDLE	LOAD	DAY
1	17	1	1/ 3/86	1	.56	3:30	0	5

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	JEFFERSON & CHURCH	10: 7: 0	10: 5: 0	19	1	18	.56		465	NO
2	JEFFERSON & FRANKLIN RD.	10: 7:30	0: 0: 0	1	0	19	.60		0	NO
4	JEFFERSON & ELM	10: 8:15	0: 0: 0	0	1	18	.78		0	NO
8	JEFFERSON & MAPLE	10: 9:45	0: 0: 0	1	0	19	1.07		0	NO
20	JEFFERSON & 23 rd	10:14:15	0: 0: 0	0	1	18	2.41		0	NO
21	JEFFERSON & 24 th	10:14:45	0: 0: 0	0	1	17	2.50		0	NO
29	26 th & AVENHAM	10:16:15	0: 0: 0	0	1	16	3.11		0	NO
41	FRANKLIN RD. & TOWNSIDE	10:20: 0	0: 0: 0	0	1	15	4.37		0	NO
43	SUPER X	10:23:15	0: 0: 0	0	4	11	5.14		0	NO
45	BRENDLES	10:32:30	10:25: 0	3	8	6	5.38		510	NO
76	JEFFERSON & YELLOW MT.	10:40:15	0: 0: 0	1	0	7	8.46		0	NO
93	JEFFERSON & CHURCH	10:49: 0	10:50: 0	3	2	8	10.17		150	NO
95	CAMPBELL & MARKET ST.	10:51:30	0: 0: 0	4	0	12	10.38		0	NO
102	GAINSBORO & PATTON	10:54:45	0: 0: 0	0	1	11	11.02		0	NO
126	STAUNTON & 22 nd	11: 1:30	0: 0: 0	0	1	10	13.23		0	NO
150	FERNCLIFF APARTMENTS N.	0: 0: 0	11:10: 0	0	0	10	13.23		0	NO
151	FERNCLIFF APARTMENTS S.	11:13: 0	0: 0: 0	4	0	14	15.82		255	NO
153	FERNCLIFF &	11:14: 0	0: 0: 0	1	0	15	16.00		0	NO
173	STAUNTON & 23 rd	11:20: 0	0: 0: 0	2	0	17	17.94		0	NO
186	STAUNTON & 11 th	11:23: 0	0: 0: 0	1	0	18	18.96		0	NO
187	11 th & HANOVER	11:24: 0	0: 0: 0	1	1	18	19.08		0	NO
188	ORANGE & 10 th	11:24:45	11:22: 0	2	0	20	19.14		0	NO
198	GAINSBORO & PATTON	11:28: 0	0: 0: 0	1	0	21	20.24	*	0	NO
199	1 st & GILMER	11:28:45	0: 0: 0	1	0	22	20.30	*	0	NO
207	SALEM & 1 st	11:31:45	0: 0: 0	0	1	21	20.85		0	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	JEFFERSON & CHURCH	11:35:45	11:35: 0	13	2	32	20.85		1515	NO
2	JEFFERSON & FRANKLIN RD.	11:36:30	0: 0: 0	1	0	33	20.90		0	NO
3	JEFFERSON & BULLIT	11:37: 0	0: 0: 0	0	1	32	20.98		0	NO
4	JEFFERSON & ELM	11:37:45	0: 0: 0	0	1	31	21.07		0	NO
7	JEFFERSON & ALBEMARLE	11:39: 0	0: 0: 0	0	1	30	21.27		0	NO
16	ROANOKE MEMORIAL HOSPITAL	11:42:30	11:42: 0	0	1	29	22.36		0	NO
29	26 th & AVENHAM	11:45:30	0: 0: 0	1	0	30	23.47		0	NO
33	AVENHAM & WILDWOOD	11:47: 0	0: 0: 0	0	1	29	23.82		0	NO
43	SUPER X	11:51:45	0: 0: 0	0	1	28	25.33		0	NO
44	PENNEY'S	11:53: 0	0: 0: 0	0	4	24	25.42		0	NO
45	BRENDLES	12: 1:15	11:55: 0	5	6	23	25.66		375	NO
49	FRANKLIN RD. & TANGLEWOOD	12: 3:45	0: 0: 0	0	1	22	26.20		0	NO
57	AVENHAM & WHITE OAK (2)	12: 6:15	0: 0: 0	1	0	23	27.12		0	NO
93	JEFFERSON & CHURCH	12:20: 0	12:20: 0	3	3	23	30.45		195	NO
94	JEFFERSON & CAMPBELL	12:21:15	0: 0: 0	5	0	28	30.53		0	NO
95	CAMPBELL & MARKET ST.	12:23:15	0: 0: 0	4	0	32	30.65		0	NO
105	GAINSBORO & McDOWELL	12:28:45	0: 0: 0	0	1	31	31.56		60	NO
111	ORANGE & 10 th	12:30:45	12:28: 0	0	1	30	32.34		0	NO
112	ORANGE & 11 th	12:32:15	0: 0: 0	0	1	29	32.46	*	0	NO
121	STAUNTON & 17 th	12:34:15	0: 0: 0	1	0	30	33.07		0	NO
122	STAUNTON & 18 th	12:34:45	0: 0: 0	0	1	29	33.15		0	NO

123	STAUNTON & 19 th	12:35:30	0: 0: 0	0	1	28	33.24	*	0	NO
124	STAUNTON & 20 th	12:36: 0	0: 0: 0	0	1	27	33.32		0	NO
130	LAFAYETTE & NEW YORK	12:37:45	0: 0: 0	0	1	26	33.88		0	NO
134	LAFAYETTE & FLORIDA	12:38:30	0: 0: 0	0	1	25	34.22		0	NO
138	COVE ROAD & ASPEN	12:39:30	0: 0: 0	0	1	24	34.59		0	NO
150	FERNCLIFF APARTMENTS N.	0: 0: 0	12:45: 0	0	0	24	34.59		0	NO
151	FERNCLIFF APARTMENTS S.	12:48:30	0: 0: 0	1	0	25	36.08		255	NO
153	FERNCLIFF &	12:49:45	0: 0: 0	2	2	25	36.27		0	NO
178	STAUNTON & 18 th	12:56:15	0: 0: 0	1	0	26	38.63		0	NO
180	STAUNTON & 16 th	12:57: 0	0: 0: 0	1	0	27	38.80		0	NO
185	STAUNTON & 12 th	12:58:15	0: 0: 0	1	0	28	39.14		0	NO
208	SALEM & JEFFERSON	13: 6:30	0: 0: 0	2	1	29	41.23		0	NO

STOP		ARRIVE	SCHED	ON	OFF	LOAD	DISTANCE	MATCH	IDLE	CHANGE?
1	JEFFERSON & CHURCH	13: 9:45	13:10: 0	5	0	34	41.24		2535	NO
2	JEFFERSON & FRANKLIN RD.	13:11: 0	0: 0: 0	2	0	36	41.27		0	NO
16	ROANOKE MEMORIAL HOSPITAL	13:16: 0	13:17: 0	1	0	37	42.73		0	NO
43	SUPER X	13:27:15	0: 0: 0	3	2	38	45.71		0	NO
44	PENNEY'S	13:29:15	0: 0: 0	3	2	39	45.80		60	NO
45	BRENDLES	13:32:30	13:30: 0	1	1	39	46.03		60	NO
46	OGDEN & FRANKLIN ROAD	13:34:30	0: 0: 0	0	1	38	46.32		0	NO
47	FRANKLIN RD. & ELM VIEW R	13:35: 0	0: 0: 0	0	1	37	46.39		0	NO
53	FRANKLIN RD. & K-MART	13:38:15	0: 0: 0	1	0	38	47.17		0	NO
57	AVENHAM & WHITE OAK (2)	13:39:30	0: 0: 0	1	0	39	47.49	*	0	NO
67	26 th & CAROLINA	13:42:15	0: 0: 0	1	0	40	48.41		0	NO
68	26 th & ROSALIND	13:43: 0	0: 0: 0	0	1	39	48.48	*	0	NO
71	26 th & JEFFERSON	13:43:45	0: 0: 0	1	0	40	48.69		0	NO
75	JEFFERSON & 22 nd	13:45: 0	0: 0: 0	0	1	39	49.03		0	NO
78	ROANOKE MEMORIAL HOSPITAL	13:46: 0	13:45: 0	1	0	40	49.33	*	0	NO
85	WALNUT AVENUE BRIDGE 1	13:49: 0	0: 0: 0	0	1	39	50.17		0	NO
86	JEFFERSON & MAPLE	13:49:45	0: 0: 0	1	0	40	50.26		0	NO
93	JEFFERSON & CHURCH	13:55:15	13:55: 0	2	3	39	50.82		195	NO
94	JEFFERSON & CAMPBELL	13:56:45	0: 0: 0	3	1	41	50.90		0	NO
95	CAMPBELL & MARKET ST.	13:58: 0	0: 0: 0	5	0	46	51.02		0	NO
102	GAINSBORO & PATTON	14: 1:45	0: 0: 0	1	0	47	51.67		0	NO
104	GAINSBORO & MADISON	14: 2:30	0: 0: 0	0	1	46	51.85		0	NO
106	GAINSBORO & ORANGE	14: 3:30	0: 0: 0	1	0	47	52.02		0	NO
118	STAUNTON & EUREKA CIRCLE	14: 7:15	0: 0: 0	0	1	46	53.15		0	NO
125	STAUNTON & 21 st	14:10: 0	0: 0: 0	0	2	44	53.70		0	NO
149	ROUTT & FERNCLIFF	14:17: 0	0: 0: 0	0	1	43	56.13		60	NO
150	FERNCLIFF APARTMENTS N.	14:18:15	14:15: 0	2	0	45	56.29		0	NO
151	FERNCLIFF APARTMENTS S.	14:19:30	0: 0: 0	5	1	49	56.45	*	0	NO
152	FERNCLIFF &	14:20: 0	0: 0: 0	2	0	51	56.56	*	0	NO
153	FERNCLIFF &	14:20:45	0: 0: 0	1	0	52	56.63		0	NO
176	STAUNTON & 20 th	14:27:15	0: 0: 0	1	0	53	58.82		0	NO
177	STAUNTON & 19 th	14:27:45	0: 0: 0	1	0	54	58.90		0	NO
178	STAUNTON & 18 th	14:28:15	0: 0: 0	0	1	53	58.99	*	0	NO
187	11 th & HANOVER	14:30: 0	0: 0: 0	0	1	52	59.66		0	NO
195	GAINSBORO & McDOWELL	14:33:30	0: 0: 0	1	0	53	60.57		0	NO
208	SALEM & JEFFERSON	14:39:30	0: 0: 0	0	1	52	61.58		60	NO

APPENDIX G. SECTION 15 REPORT PROGRAM

```
C ***************************************************************************
C     PROGRAM NAME : SECTION 15 REPORT PROGRAM
C ***************************************************************************
C
C     Written By : UMESH D. AVADHANI
C     Date       : NOVEMBER 9, 1985
C
C     THIS PROGRAM READS A MATRIX OF DATA, WHICH IS OBTAINED FROM THE
C     FIRST STAGE OF THIS EXPERIMENTAL PROJECT. THE SECTION 15 REPORT
C     IS SUBMITTED TO THE UMTA ANNUALLY BY THE ROANOKE VALLEY METRO .
C
C     THE VARIOUS VARIABLES USED IN THIS PROGRAM ARE DEFINED IN THE
C     APPENDIX OF THE THESIS.
C
C     UNIT 5 = TEST.OUT ( the input file obtained from 1st stage )
C     UNIT 6 = SECTT.OUT (the output of this program is stored in this
C                         file -- SECTION 15 REPORT )
C
      COMMON /DATA1/ IARRV, IONS, IOFFS, LOAD, IDIST, IP,STN
      COMMON /VALUES/ VALUE, SALTO, LUCKY, DEPTE, AMDIST
      COMMON /DISTAS/ DISTA1, DISTA2, DISTA3, DISTA4
      COMMON /LEFTS/ LEFT1, LEFT2, LEFT3, LEFT4
      COMMON /MILES/ ISEAMI,ICAPMI,TVEHHR,TSEAMI,TCAPMI,TLOAD,TVEHMI
      COMMON /PASSS/ PASS1, PASS2, PASS3, PASS4
      COMMON /ROUTE/ RTENO, B, LIMITA, TPASMI, APASS
      COMMON /ONS/ IPASSN, IPASSF, ILOAD, HRS
      COMMON /DATES/ IDATE, YEAR, MO
      COMMON /SATVAL/ SVALUE, SATLEF, SATPAS, SATAP, ISA
      COMMON /SATONN/ SATON, SATLOD, SATDIS
      COMMON /SATMIL/ SAPSMI, SACPTY, SASEMI, SAVEHM
      COMMON /SATANL/ SUPPT, SPMPT, SUPTAT, SPMAT
      COMMON /UNLINK/ UPPT, PMPT, UPTAT, PMAT
      COMMON /INPUT1/ IAM, ISAT,IS, LL
      COMMON /INPUT2/ IT, ITT, IA, IW, IX, IZ
      COMMON /ANNUAL/ IAPASN, IALOAD, IAVHMI, IAPSMI, IACAPM, IASEMI
C
      INTEGER IARRV(300), IONS(300), IOFFS(300), LOAD(300), IDIST(300)
      INTEGER RTENO(15), DAY(60), B, SUN, N, IDATE, YEAR, MO, LL, IZ(4)
      INTEGER DISTA1(200),DISTA2(200),DISTA3(200),DISTA4(200),
     * VALUE, SALTO, LUCKY, DEPTE,LEFT1(200),LEFT2(200),LEFT3(200)
      INTEGER IPASSN(4),IPASSF(4),ILOAD(4),HRS(4),LEFT4(200)
      INTEGER ISEAMI(4),ICAPMI(4),AMDIST(4),IX(4), LIMITA, IP(5),IAM(4)
      INTEGER TCAPMI(4),TLOAD(4),TSEAMI(4),TVEHMI(4),TVEHHR(4)
      INTEGER APASS(4),PASS1(200),PASS2(200),PASS3(200),PASS4(200)
      INTEGER SVALUE, SATLEF, SATPAS, SATAP
      INTEGER SATON, SATLOD, SATDIS, TPASMI(4)
      INTEGER SAPSMI, SACPTY, SASEMI, SAVEHM
      INTEGER SUPPT, SPMPT, SUPTAT, SPMAT
```

```
      INTEGER UPPT(4), PMPT(4), UPTAT(4), PMAT(4),STN
      CHARACTER*8 DATES
      DATA  CAPCTY,SEATS,VEHCL/60.0,40.0,1.0/
C
1002  OPEN(5, FILE = '  ')
C
      IB = 0
      SUN = 0
C     IA = 1
      B = 1
      RTENO(0) = 0
      DAY(0) = 0
4000  READ(5,1,END=6000) RTENO(B), DAY(B)
1     FORMAT(1X,I4,28X,I3)
      IF (DAY(B) .NE. DAY(B-1)) THEN
         IF (DAY(0) .EQ. 0) THEN
            DAY(0) = 1
            IA = IA + 1
            GO TO 3002
         ENDIF
       IA = IA + 1
      ENDIF
C
C     IF (DAY(IA) .EQ. 6) GO TO 3001
      IF (RTENO(B) .NE. RTENO(B-1)) THEN
          GO TO 2000
      ELSE
          GO TO 1000
      ENDIF
C
2000  CALL ROUTES
C
1000  IF (DAY(IA) .NE. 6) THEN
            CALL WKDAY
            GO TO 4000
      ELSE
            CALL STRDAY
      ENDIF
C
6000  IT = IX(1)+IX(2)+IX(3)+IX(4)+IB
C
      WRITE (*,*) 'ANY MORE DATA TO BE PROCESSED, YES = 1, NO = 0 '
      READ (*,*) MM
      IF (MM .EQ. 1) GO TO 1002
C
      CALL INPUT
C
      OPEN(6, FILE = '  ',STATUS = 'NEW')
C
      WRITE(6,11) MO,IDATE,YEAR
```

```
      WRITE(*,11) MO,IDATE,YEAR
C
      WRITE (6,12)
      WRITE (*,12)
C
      WRITE(6,10) IPASSN(1),IPASSN(2),IPASSN(3),IPASSN(4),SATON,SUN,
     *            IAPASN
      WRITE(*,10) IPASSN(1),IPASSN(2),IPASSN(3),IPASSN(4),SATON,SUN,
     *            IAPASN
      WRITE(6,20) ILOAD(1),ILOAD(2),ILOAD(3),ILOAD(4),SATLOD,SUN,IALOD
      WRITE(*,20) ILOAD(1),ILOAD(2),ILOAD(3),ILOAD(4),SATLOD,SUN,IALOD
      WRITE(6,30) TVEHMI(1),TVEHMI(2),TVEHMI(3),TVEHMI(4),SAVEHM,SUN,
     *            IAVHMI
      WRITE(*,30) TVEHMI(1),TVEHMI(2),TVEHMI(3),TVEHMI(4),SAVEHM,SUN,
     *            IAVHMI
      WRITE(6,40) TPASMI(1),TPASMI(2),TPASMI(3),TPASMI(4),SAPSMI,SUN,
     *            IAPSMI
      WRITE(*,40) TPASMI(1),TPASMI(2),TPASMI(3),TPASMI(4),SAPSMI,SUN,
     *            IAPSMI
      WRITE(6,50) TCAPMI(1),TCAPMI(2),TCAPMI(3),TCAPMI(4),SACPTY,SUN,
     *            IACAPM
      WRITE(*,50) TCAPMI(1),TCAPMI(2),TCAPMI(3),TCAPMI(4),SACPTY,SUN,
     *            IACAPM
      WRITE(6,60) TSEAMI(1),TSEAMI(2),TSEAMI(3),TSEAMI(4),SASEMI,SUN,
     *            IASEMI
      WRITE(*,60) TSEAMI(1),TSEAMI(2),TSEAMI(3),TSEAMI(4),SASEMI,SUN,
     *            IASEMI
      WRITE(6,70) IZ(1),IZ(2),IZ(3),IZ(4),ISA,SUN,IT
      WRITE(*,70) IZ(1),IZ(2),IZ(3),IZ(4),ISA,SUN,IT
      WRITE(6,80) IAM(1),IAM(2),IAM(3),IAM(4),ISAT,SUN,ITT
      WRITE(*,80) IAM(1),IAM(2),IAM(3),IAM(4),ISAT,SUN,ITT
      WRITE(6,81)
      WRITE(*,81)
      WRITE(6,82) UPPT(1),UPPT(2),UPPT(3),UPPT(4),SUPPT,SUN
      WRITE(*,82) UPPT(1),UPPT(2),UPPT(3),UPPT(4),SUPPT,SUN
      WRITE(6,83) PMPT(1),PMPT(2),PMPT(3),PMPT(4),SPMPT,SUN
      WRITE(*,83) PMPT(1),PMPT(2),PMPT(3),PMPT(4),SPMPT,SUN
      WRITE(6,84)
      WRITE(*,84)
      WRITE(6,85) UPTAT(1),UPTAT(2),UPTAT(3),UPTAT(4),SUPTAT,SUN
      WRITE(*,85) UPTAT(1),UPTAT(2),UPTAT(3),UPTAT(4),SUPTAT,SUN
      WRITE(6,86) PMAT(1),PMAT(2),PMAT(3),PMAT(4),SPMAT,SUN
      WRITE(*,86) PMAT(1),PMAT(2),PMAT(3),PMAT(4),SPMAT,SUN
C
11    FORMAT(47X,' Form 406A ',/,42X,' ANNUAL REPORT TO UMTA ',/,/,
     * ' Transit Id 3007 ',30X,' Non-Rail Modes ',/,
     * ' Fiscal Year End ',
     * 1X,I2,'/',I2,'/',I2, 71X,' Mode  MB ',/,97X,
     * ' Level  R ',/,53X, ' WEEKDAYS ',/,' Line ',100X,' ANNUAL ',
     * /,1X,' no. ', 6X, ' ITEM ',17X,' AM PEAK ', 4X, ' MIDDAY ', 4X,
     * ' PM PEAK ', 4X, ' OTHER ',2X,' SATURDAY ',2X,' SUNDAY ',5X,
```

```
     * ' TOTAL ',/)
12    FORMAT(7X,' Accumulation From ',/,7X,' Daily Record Sheet ',/)
10    FORMAT(2X,'01',3X,' Passengers Boarded ',9X,I7,7X,I5,
     * 8X,I5,6X,I5,7X,I5,5X,I5,5X,I5,/)
20    FORMAT(2X,'02',3X,' Passengers on Board ',8X,I7,7X,I5,8X,
     * I5,6X,I5,7X,I5,5X,I5,5X,I5,/)
30    FORMAT(2X,'03',3X,' Bus Trip Distance ',10X,I7,7X,I5,8X,I5,6X,I5,
     * 7X,I5,5X,I5,5X,I5,/)
40    FORMAT(2X,'04',3X,' Passenger Miles ',12X,I7,7X,I5,8X,I5,
     * 6X,I5,7X,I5,5X,I5,5X,I5,/)
50    FORMAT(2X,'05',3X,' Capacity Miles ',13X,I7,7X,I5,8X,I5,6X,I5,
     * 7X,I5,5X,I5,5X,I5/)
60    FORMAT(2X,'06',3X,' Seat Miles ',17X,I7,7X,I5,8X,I5,6X,I5,
     * 7X,I5,5X,I5,5X,I5/)
70    FORMAT(2X,'07',3X,' Trips in Sample ',12X,I7,7X,I5,8X,I5,6X,I5,
     * 7X,I5,5X,I5,5X,I5,/)
80    FORMAT(2X,'08',3X,' Total Number of Bus Trips ',
     * 2X,I7,7X,I5,8X,I5,6X,I5,7X,I5,5X,I5,5X,I5//)
81    FORMAT(7X,' Sample Averages ',/)
82    FORMAT(2X,'09',3X,' Unlinked Pass. per Trip (1/7)',1X,I5,7X,  ,
     * I5, 8X,I5,6X,I5,7X,I5,5X,I5,/)
83    FORMAT(2X,'10',3X,' Pass. Miles per Trip (4/7)',3X,I6,7X,I5,
     *  8X,I5,6X,I5,7X,I5,5X,I5,//)
84    FORMAT(7X,' Annual Totals ',/)
85    FORMAT(2X,'11',3X,' Unlinked Passenger Trips (8x9)',3X,I2,7X,I5,
     *  8X,I5,6X,I5,7X,I5,5X,I5,/)
86    FORMAT(2X,'12',3X,' Passenger Miles (8x10)',6X,I7,7X,I5,
     *  8X,I5,6X,I5,7X,I5,5X,I5,/)
C
  99  END
C
C
C     SUBROUTINE ROUTES
C
      SUBROUTINE ROUTES
      COMMON /ROUTE/ RTENO, B, LIMITA, TPASMI, APASS
      INTEGER RTENO(15), B, LIMITA, APASS(4), TPASMI(4)
C
      IF (RTENO(B) .EQ. 1) THEN
        WRITE(*,*) 'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * ' COVE ROAD/PROSPECT HILLS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 2) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'LINCOLN TERRACE/SALEM-SHENANDOAH ROUTE'
        READ(*,*) LIMITA
      B = B + 1
```

```
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 3) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'RIVERDALE/RUGBY ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 4) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'KENWOOD/VINTON-WISE AVENUE ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 5) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'MELROSE/VIRGINIA HEIGHTS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 6) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'HUNTINGTON COURT/WASENA ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 7) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'RALEIGH COURT/VILLA HEIGHTS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 8) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'COLONIAL HEIGHTS/WILLIAMSON ROAD ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
```

```
      IF (RTENO(B) .EQ. 9) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'GRANDIN COURT ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
C
      IF (RTENO(B) .EQ. 10) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'EAST ORANGE AVENUE ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 5000
      ENDIF
5000  RETURN
      END
C
C
C     SUBROUTINE WKDAY
C
      SUBROUTINE WKDAY
      COMMON /DATA1/ IARRV, IONS, IOFFS, LOAD, IDIST, IP,STN
      COMMON /VALUES/ VALUE, SALTO, LUCKY, DEPTE, AMDIST
      COMMON /DISTAS/ DISTA1, DISTA2, DISTA3, DISTA4
      COMMON /LEFTS/ LEFT1, LEFT2, LEFT3, LEFT4
      COMMON /MILES/ ISEAMI,ICAPMI,TVEHHR,TSEAMI,TCAPMI,TLOAD,TVEHMI
      COMMON /PASSS/ PASS1, PASS2, PASS3, PASS4
      COMMON /ROUTE/ RTENO, B, LIMITA, TPASMI, APASS
      COMMON /ONS/ IPASSN, IPASSF, ILOAD, HRS
      COMMON /INPUT2/ IT, ITT, IA, IW, IX, IZ
C
      INTEGER IARRV(300),IONS(300),IOFFS(300),LOAD(300),IDIST(300),STN
      INTEGER DISTA1(200),DISTA2(200),DISTA3(200),DISTA4(200),RTENO(15),
     * VALUE, SALTO, LUCKY, DEPTE,LEFT1(200),LEFT2(200),LEFT3(200)
      INTEGER IPASSN(4),IPASSF(4),ILOAD(4),HRS(4),LEFT4(200),TPASMI(4)
      INTEGER ISEAMI(4),ICAPMI(4),AMDIST(4),IX(4), LIMITA, IP(5),B
      INTEGER TCAPMI(4),TLOAD(4),TSEAMI(4),TVEHMI(4),TVEHHR(4)
      INTEGER APASS(4),PASS1(200),PASS2(200),PASS3(200),PASS4(200)
      INTEGER IT, ITT, IA, IW, IZ(4)
C
      DO 100 I = 1, LIMITA + 1
C
            IF (I .EQ. LIMITA) GO TO 100
C
            IF (I .EQ. LIMITA + 1) GO TO 3000
C
      READ(5,5,END=3000) IARRV(I),IONS(I),IOFFS(I),LOAD(I),IDIST(I)
5     FORMAT(35X,I6,10X,I2,2X,I2,1X,I3,3X,I8)
       J=1
       K=1
```

```
            IF (IARRV(I).GE.21600.AND.IARRV(I).LT.32400) THEN
             IF(I .EQ. LIMITA-1) IP(J) = IP(J)+2
               IX(J) = IX(J) + 1
               DISTA1(IX(J)) = IDIST(I)
               VALUE = DISTA1(IX(J)) - DISTA1(1)
               LEFT1(IX(J)) = DISTA1(IX(J)) - IDIST(I-1)
               PASS1(IX(J)) = LOAD(I-1) * LEFT1(IX(J))
               APASS(K) = PASS1(IX(J)) + APASS(K)
C
               IPASSN(J) = IONS(I) + IPASSN(J)
               ILOAD(J) = LOAD(I) + ILOAD(J)
            ENDIF
      AMDIST(J) = FLOAT(VALUE)/FLOAT(5280)
      J=2
      K=2
            IF (IARRV(I).GE.32400.AND.IARRV(I).LT.54000) THEN
             IF(I .EQ. LIMITA-1) IP(J) = IP(J)+2
               IX(J) = IX(J) + 1
               DISTA2(IX(J)) = IDIST(I)
               SALTO = DISTA2(IX(J)) - DISTA2(1)
               LEFT2(IX(J)) = DISTA2(IX(J)) - IDIST(I-1)
               PASS2(IX(J)) = LOAD(I-1) * LEFT2(IX(J))
               APASS(K) = PASS2(IX(J)) + APASS(K)
C
               IPASSN(J) = IONS(I) + IPASSN(J)
               ILOAD(J) = LOAD(I) + ILOAD(J)
          ENDIF
      AMDIST(J) = FLOAT(SALTO)/FLOAT(5280)
      J=3
      K=3
            IF (IARRV(I).GE.54000.AND.IARRV(I).LT.64800) THEN
             IF(I .EQ. LIMITA-1) IP(J) = IP(J)+2
               IX(J) = IX(J) + 1
               DISTA3(IX(J)) = IDIST(I)
               LUCKY = DISTA3(IX(J)) - DISTA3(1)
               LEFT3(IX(J)) = DISTA3(IX(J)) - IDIST(I-1)
               PASS3(IX(J)) = LOAD(I-1) * LEFT3(IX(J))
               APASS(K) = PASS3(IX(J)) + APASS(K)
C
               IPASSN(J) = IONS(I) + IPASSN(J)
               ILOAD(J) = LOAD(I) + ILOAD(J)
            ENDIF
      AMDIST(J) = FLOAT(LUCKY)/FLOAT(5280)
      J=4
      K=4
            IF (IARRV(I).GE.64800.AND.IARRV(I).LE.82800) THEN
             IF(I .EQ. LIMITA-1) IP(J) = IP(J)+2
               IX(J)=IX(J) + 1
               DISTA4(IX(J)) = IDIST(I)
               DEPTE = DISTA4(IX(J)) - DISTA4(1)
               LEFT4(IX(J)) = DISTA4(IX(J)) - IDIST(I-1)
```

```
                PASS4(IX(J)) = LOAD(I-1) * LEFT4(IX(J))
                APASS(K) = PASS4(IX(J)) + APASS(K)
C
                IPASSN(J) = IONS(I) + IPASSN(J)
                ILOAD(J) = LOAD(I) + ILOAD(J)
            ENDIF
      AMDIST(J) = FLOAT(DEPTE)/FLOAT(5280)
100   CONTINUE
3000  READ(5,*) STN
      RETURN
      END
C
C     SUBROUTINE STRDAY
C
      SUBROUTINE STRDAY
      COMMON /DATA1/ IARRV, IONS, IOFFS, LOAD, IDIST,IP,STN
      COMMON /ROUTE/ RTENO, B, LIMITA, TPASMI, APASS
      COMMON /SATVAL/ SVALUE, SATLEF, SATPAS, SATAP, ISA
      COMMON /SATONN/ SATON, SATLOD, SATDIS
      COMMON /SATMIL/ SAPSMI, SACPTY, SASEMI, SAVEHM
      COMMON /SATANL/ SUPPT, SPMPT, SUPTAT, SPMAT
C
      INTEGER IARRV(300), IONS(300), IOFFS(300), LOAD(300), IDIST(300)
      INTEGER SVALUE, SATLEF, SATPAS, SATAP, TPASMI(4)
      INTEGER SATON, SATLOD, SATDIS,B,LIMITA,RTENO(15)
      INTEGER SAPSMI, SACPTY, SASEMI, SAVEHM, APASS(4)
      INTEGER SUPPT, SPMPT, SUPTAT, SPMAT, IP(5),STN

      ISA = ISA + 1

      DO 700 N = 1, LIMITA + 1
C
            IF (I .EQ. LIMITA) GO TO 700
C
            IF (I .EQ. LIMITA + 1) GO TO 3000
C
      READ(5,5,END=3000) IARRV(N),IONS(N),IOFFS(N),LOAD(N),IDIST(N)
5     FORMAT(35X,I6,10X,I2,2X,I2,1X,I3,3X,I8)
              IB = IB + 1
             IF(I .EQ. LIMITA-1) IP(5) = IP(5)+2
C
      SVALUE = IDIST(N) - IDIST(1)
      SATLEF = IDIST(N) - IDIST(N-1)
      SATPAS = LOAD(N-1) * SATLEF
      SATAP = SATPAS + SATAP
C
      SATON = IONS(N) + SATON
      SATLOD = LOAD(N) + SATLOD
      SATDIS = FLOAT(SVALUE)/FLOAT(5280)
700   CONTINUE
C
```

```
          SAPSMI = FLOAT(SATAP)/FLOAT(5280)
          SACPTY = SATDIS * CAPCTY
          SASEMI = SATDIS * SEATS
          SAVEHM = SATDIS * VEHCL
          SUPPT = SATON/IP(5)
          SPMPT = SAPSMI/IP(5)
          SUPTAT = SPMPT * ISAT
          SPMAT = SPMPT * ISAT
C
3000  READ(5,*) STN
      RETURN
      END
C
C
C     SUBROUTINE INPUT
C
      SUBROUTINE INPUT
      COMMON /VALUES/ VALUE, SALTO, LUCKY, DEPTE, AMDIST
      COMMON /MILES/ ISEAMI,ICAPMI,TVEHHR,TSEAMI,TCAPMI,TLOAD,TVEHMI
      COMMON /PASSS/ PASS1, PASS2, PASS3, PASS4
      COMMON /ROUTE/ RTENO, B, LIMITA, TPASMI, APASS
      COMMON /ONS/ IPASSN, IPASSF, ILOAD, HRS
      COMMON /DATES/ IDATE, YEAR, MO
      COMMON /SATVAL/ SVALUE, SATLEF, SATPAS, SATAP, ISA
      COMMON /SATONN/ SATON, SATLOD, SATDIS
      COMMON /SATMIL/ SAPSMI, SACPTY, SASEMI, SAVEHM
      COMMON /SATANL/ SUPPT, SPMPT, SUPTAT, SPMAT
      COMMON /UNLINK/ UPPT, PMPT, UPTAT, PMAT
      COMMON /INPUT1/ IAM, ISAT,IS, LL
      COMMON /INPUT2/ IT, ITT, IA, IW, IX, IZ
      COMMON /ANNUAL/ IAPASN, IALOAD, IAVHMI, IAPSMI, IACAPM, IASEMI
C
      INTEGER SVALUE, SATLEF, SATPAS, SATAP, IAM(4)
      INTEGER SATON, SATLOD, SATDIS, LIMITA, B, RTENO(15)
      INTEGER UPPT(4), PMPT(4), UPTAT(4), PMAT(4)
      INTEGER SAPSMI, SACPTY, SASEMI, SAVEHM
      INTEGER TCAPMI(4),TLOAD(4),TSEAMI(4),TVEHMI(4),TVEHHR(4),TPASMI(4)
      INTEGER IPASSN(4),IPASSF(4),ILOAD(4),HRS(4),LEFT4(200)
      INTEGER AMDIST(4), APASS(4), PASS1(200),PASS2(200),PASS3(200)
      INTEGER IDATE, YEAR, MO, PASS4(200), LL, IZ(4)
      INTEGER ISEAMI(4),ICAPMI(4),IX(4), IP(5)
      DATA  CAPCTY, SEATS, VEHCL /60.0,40.0,1.0/
C
      WRITE(*,*) IPASSN(3),ILOAD(3)
C
      WRITE(*,*) IPASSN(4),ILOAD(4)
C
      WRITE(*,*) 'PLEASE ENTER THE TOTAL NUMBER OF TRIPS DURING THE AM
     * PEAK = '
      READ(*,*) IAM(1)
      WRITE(*,*) 'MIDDAY = '
```

```
      READ(*,*) IAM(2)
      WRITE(*,*) 'PM PEAK = '
      READ(*,*) IAM(3)
      WRITE(*,*) 'OTHER = '
      READ(*,*) IAM(4)
      WRITE(*,*) 'SATURDAY = '
      READ(*,*) ISAT
C
C     IW = IAM(1)+IAM(2)+IAM(3)+IAM(4)
      WRITE(*,*) 'PLEASE ENTER THE TOTAL NUMBER OF WEEKDAYS THAT SERVICE
     * WAS OPERATED'
      READ(*,*) IW
C
      WRITE(*,*)'PLEASE ENTER THE TOTAL NUMBER OF SATURDAYS THAT SERVICE
     * WAS OPERATED'
      READ(*,*) IS
      ITT = IW + IS
C
        DO 121 I = 1, 4
           IPASSN(I) = IPASSN(I)/IA
           ILOAD(I) = ILOAD(I)/IA
           TPASMI(I) = APASS(I)/(5280.00*IA)
           TCAPMI(I) = AMDIST(I) * CAPCTY/IA
           TSEAMI(I) = AMDIST(I) * SEATS/IA
           TVEHMI(I) = AMDIST(I) * VEHCL/IA
           UPPT(I) = IPASSN(I)/IZ(I)
           PMPT(I) = TPASMI(I)/IZ(I)
           UPTAT(I) = UPPT(I) * IAM(I)
           PMAT(I) = PMPT(I) * IAM(I)
121     CONTINUE
C
      IAPASN = (IPASSN(1)+IPASSN(2)+IPASSN(3)+IPASSN(4))*IW+SATON*IS
      IALOAD = (ILOAD(1)+ILOAD(2)+ILOAD(3)+ILOAD(4))*IW+SATLOD*IS
      IAVHMI = (TVEHMI(1)+TVEHMI(2)+TVEHMI(3)+TVEHMI(4))*IW+SAVEHM*IS
      IAPSMI = (TPASMI(1)+TPASMI(2)+TPASMI(3)+TPASMI(4))*IW+SAPSMI*IS
      IACAPM = (TCAPMI(1)+TCAPMI(2)+TCAPMI(3)+TCAPMI(4))*IW+SACPTY*IS
      IASEMI = (TSEAMI(1)+TSEAMI(2)+TSEAMI(3)+TSEAMI(4))*IW+SASEMI*IS
C
      WRITE (*,*)  'WHEN DOES THE FISCAL YEAR END? , MONTH = '
      READ (*,*) MO
      WRITE (*,*) 'DAY = '
      READ (*,*) IDATE
      WRITE (*,*) 'YEAR = '
      READ (*,*) YEAR
      RETURN
      END
```

APPENDIX H. TIME POINT TRIP FILE PROGRAM

```
C  ***************************************************************************
C     PROGRAM NAME : TRIP POINT TIME FILE PROGRAM
C  ***************************************************************************
C
C     Written By : UMESH D. AVADHANI
C     Date       : NOVEMBER 9, 1985
C
C     Purpose    : READS THE OUTPUT FROM THE FIRST STAGE OF THIS
C                  PROJECT AND THEN AFTER CERTAIN SUITABLE
C                  CALCULATIONS GIVES OUT THE TRIP POINT TIME FILE
C
C     UNIT 5 = TEST.OUT (the input to derive the TIME POINT TRIP FILE )
C     UNIT 6 = TRIP.OUT (the output of this program )
C     UNIT 7 = PROFILE.OUT (this output is used as an input to create
C                           the LOAD PROFILE PLOT on SYMPHONY )
C
      COMMON /TIME/ IARRV, ISCH,IDEP,ISCHDV, IMINN
      COMMON /DATA1/ IONS, IOFFS,LOAD, ISTN, IN, INSTN
      COMMON /DATA2/ RTENO, BLKNO, MO, DATE, YEAR
      COMMON /SORT/ MARR, BARRV, DAY, PASSON, PASOFF, L, K, LARRV
      COMMON /STNS/ STN, BSTN, N, LSTN, MSTN, LINN, BINN, MINN, B, A, M
      COMMON /HOURS/ IH1,IH2,IH3,IM1,IM2,IM3,IS1,IS2,IS3,LIMITA,IX,IB,D
C
      INTEGER IARRV(300), ISCH(300),IDEP(300),ISCHDV(300),IMINN(50)
      INTEGER IONS(300), IOFFS(300), LOAD(300), ISTN(300), INSTN(50)
      INTEGER PASSON(100), PASOFF(100),D
      INTEGER MARR(50), MSTN(50), DAY(10), L, K, STN, M, IN
      INTEGER LINN, LSTN, N(300), LARRV, MINN(50), B, A
      INTEGER RTENO(10), BLKNO, MO, DATE,YEAR,BARRV,BINN,BSTN
      INTEGER IH1,IH2,IH3,IM1,IM2,IM3,IS1,IS2,IS3,LIMITA,IX,IB
      CHARACTER*30 NAMES(300),NAM1(300),NNAME, NNAM1(50)
C
      OPEN(5, FILE = ' ')
      OPEN(6, FILE = ' ', STATUS = 'NEW')
      OPEN(7, FILE = ' ', STATUS = 'NEW')
      IX = 1
      PASSON(1) = 0
      PASOFF(1) = 0
      IB = 2
      N(1) = 0
C
      RTENO(0) = 0
      DAY(0) = 0
C
      WRITE (6,40)
      WRITE (*,40)
C
```

```
      WRITE (6,70)
      WRITE (*,70)
C
3000  READ(5,2,END = 99) RTENO(B), BLKNO, MO, DATE, YEAR, DAY(B)
  2   FORMAT(3X, I2, 5X, I3, 4X, I2, 4X, I2, 2X, I2, 5X, I2)
      IN = 0
      IF (DAY(B) .NE. DAY(B-1)) IA = IA + 1
      IF (DAY(B) .EQ. 6) GO TO 3001
3001  IF (RTENO(B) .NE. RTENO(B-1)) THEN
      WRITE (6,91) RTENO(B), BLKNO, MO, DATE, YEAR
      WRITE (*,91) RTENO(B), BLKNO, MO, DATE, YEAR
          GO TO 2000
      ELSE
          GO TO 1000
      ENDIF
C
2000  IF (RTENO(B) .EQ. 1) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * ' COVE ROAD/PROSPECT HILLS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 2) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'LINCOLN TERRACE/SALEM-SHENANDOAH ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 3) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'RIVERDALE/RUGBY ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 4) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'KENWOOD/VINTON-WISE AVENUE ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 5) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'MELROSE/VIRGINIA HEIGHTS ROUTE'
```

```
        READ(*,*) LIMITA
       B = B + 1
       GO TO 1000
       ENDIF
C
       IF (RTENO(B) .EQ. 6) THEN
         WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'HUNTINGTON COURT/WASENA ROUTE'
         READ(*,*) LIMITA
       B = B + 1
       GO TO 1000
       ENDIF
C
       IF (RTENO(B) .EQ. 7) THEN
         WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'RALEIGH COURT/VILLA HEIGHTS ROUTE'
         READ(*,*) LIMITA
       B = B + 1
       GO TO 1000
       ENDIF
C
       IF (RTENO(B) .EQ. 8) THEN
         WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'COLONIAL HEIGHTS/WILLIAMSON ROAD ROUTE'
         READ(*,*) LIMITA
       B = B + 1
       GO TO 1000
       ENDIF
C
       IF (RTENO(B) .EQ. 9) THEN
         WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'GRANDIN COURT ROUTE'
         READ(*,*) LIMITA
       B = B + 1
       GO TO 1000
       ENDIF
C
       IF (RTENO(B) .EQ. 10) THEN
         WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'EAST ORANGE AVENUE ROUTE'
         READ(*,*) LIMITA
       B = B + 1
       GO TO 1000
       ENDIF
C
1000   WRITE (6,80)
       WRITE (*,80)
C
       DO 100 I = 1 , LIMITA + 1
C
        IF (I .EQ. LIMITA) THEN
```

```
            D = I
            CALL MXLOAD
            GO TO 100
         ENDIF
         IF (I .EQ. LIMITA + 1) THEN
            READ(5,*) STN
            WRITE (6,1)
            WRITE (*,1)
   1        FORMAT (/)
            GO TO 3003
         ENDIF
C
       READ(5,10,END=99) ISTN(I), NAMES(I),IARRV(I),ISCH(I), IONS(I),
     *                   IOFFS(I),LOAD(I)
C
       PASSON(IX) = IONS(I) + PASSON(IX)
       PASOFF(IX) = IOFFS(I) + PASOFF(IX)
C
       IF (IARRV(I).NE.0.AND.ISCH(I).NE.0) THEN
           ISCHDV(I)=ISCH(I) - IARRV(I)
C
            IF (ISCHDV(I).LT.0) THEN
               ISCHDV(I) = (-1)*ISCHDV(I)
C
               IH1 = IARRV(I)/3600
               IH2 = ISCH(I)/3600
               IH3 = ISCHDV(I)/3600
               IM1 = (IARRV(I) - IH1*3600)/60
               IM2 = (ISCH(I) - IH2*3600)/60
               IM3 = (ISCHDV(I) - IH3*3600)/60
               IS1 = IARRV(I) - IH1*3600 - IM1*60
               IS2 = ISCH(I) - IH2*3600 - IM2*60
               IS3 = ISCHDV(I) - IH3*3600 - IM3*60
C
            WRITE(*,90) ISTN(I), NAMES(I), IH1, IM1, IS1, IH2, IM2, IS2,
     *                  IH3, IM3, IS3, PASSON(IX), PASOFF(IX),LOAD(I)
C
            WRITE(6,90) ISTN(I), NAMES(I), IH1, IM1, IS1, IH2, IM2, IS2,
     *                  IH3, IM3, IS3, PASSON(IX), PASOFF(IX),LOAD(I)
C
       WRITE(7,110) ISTN(I), NAMES(I),PASSON(IX),PASOFF(IX),LOAD(I)
110    FORMAT(2X,I3,3X,A27,2X,I4,2X,I4,2X,I4)
           ELSE
              IH1 = IARRV(I)/3600
              IH2 = ISCH(I)/3600
              IH3 = ISCHDV(I)/3600
              IM1 = (IARRV(I) - IH1*3600)/60
              IM2 = (ISCH(I) - IH2*3600)/60
              IM3 = (ISCHDV(I) - IH3*3600)/60
              IS1 = IARRV(I) - IH1*3600 - IM1*60
              IS2 = ISCH(I) - IH2*3600 - IM2*60
```

```
               IS3 = ISCHDV(I) - IH3*3600 - IM3*60
C
            WRITE(*,20) ISTN(I), NAMES(I),IH1, IM1, IS1, IH2, IM2, IS2,
     *                  IH3, IM3, IS3, PASSON(IX), PASOFF(IX),LOAD(I)
C
            WRITE(6,20) ISTN(I), NAMES(I),IH1, IM1, IS1, IH2, IM2, IS2,
     *                  IH3, IM3, IS3, PASSON(IX), PASOFF(IX),LOAD(I)
C
          WRITE(7,110) ISTN(I), NAMES(I),PASSON(IX),PASOFF(IX),LOAD(I)
C
              ENDIF
           IX = IX + 1
           GO TO 400
         ENDIF
           IF(ISCH(I).NE.0) THEN
              IH2 = ISCH(I)/3600
              IM2 = (ISCH(I) - IH2*3600)/60
              IS2 = ISCH(I) - IH2*3600 - IM2*60
            WRITE(*,30)ISTN(I),NAMES(I),IH2,IM2,IS2,PASSON(IX),PASOFF(IX),
     *    LOAD(I)
C
            WRITE(6,30)ISTN(I),NAMES(I),IH2,IM2,IS2,PASSON(IX),PASOFF(IX),
     *    LOAD(I)
C
          WRITE(7,110) ISTN(I), NAMES(I),PASSON(IX),PASOFF(IX),LOAD(I)
C
           ENDIF
C
400        IF(ISCH(I).NE.0) THEN
               D = I
               CALL MXLOAD
           ENDIF
100     CONTINUE
C
3003    DO 3004 IM = 1, IN
           WRITE(6,11) INSTN(IM), NNAM1(IM), IMINN(IM)
           WRITE(*,11) INSTN(IM), NNAM1(IM), IMINN(IM)
11         FORMAT(5X,'STOP NO. = ',I3,3X,A26,4X,'MAX. LOAD = ',I3)

3004    CONTINUE
            WRITE (6,1)
            WRITE (6,80)
            GO TO 3000
C
   10 FORMAT(1X,I3,4X,A27,1X,I5,1X,I5,3X,I3,1X,I3,1X,I3)
C
   20 FORMAT(1X,I3,4X,A26,6X,I2,':',I2,':',I2,5X,I2,':',I2,':',I2,
     *       5X,I2,':',I2,':',I2,6X,I4,4X,I4,16X, I4)
C
   30 FORMAT(1X,I3,4X,A26,19X,I2,':',I2,':',I2,19X,I4,4X,I4,16X,I4)
C
```

```
   40 FORMAT(26X,'ROANOKE VALLEY-METRO APC SYSTEM',//,
     *       31X,'TIME POINT-TRIP FILE',/)
C
   50 FORMAT(5X,I2,5X,I2,11X,I1,8X,A8,5X,I1,8X,I5,5X,I4)
C
   60 FORMAT(5X,I2,5X,I2,11X,I1,8X,A8,5X,I1,/)
C
   70 FORMAT(14X, 'ROUTE', 2X, 'BLOCK', 4X, 'DATE',/)
C
   91 FORMAT(14X, I3, 6X, I2, 3X, I2, '/', I2, '/', I2)
C
   80 FORMAT(/,1X,'STOP #',34X,'ARRIVE',7X,'SCHED',7X,'SCHDEV',7X,
     * 'PASS. BET. TIME PTS.',4X,'LOAD',/,81X,'ONS',5X,'OFFS',/)
C
   90 FORMAT(1X,I3,4X,A26,6X,I2,':',I2,':',I2,5X,I2,':',I2,':',
     *       I2,4X,'-',I2,':',I2,':',I2,6X,I4,4X,I4,16X,I4)
C
   99 END
C
C     SUBROUTINE MXLOAD
C
      SUBROUTINE MXLOAD
C
      COMMON /TIME/ IARRV, ISCH,IDEP,ISCHDV, IMINN
      COMMON /DATA1/ IONS, IOFFS,LOAD, ISTN, IN,INSTN
      COMMON /DATA2/ RTENO, BLKNO, MO, DATE, YEAR
      COMMON /SORT/ MARR, BARRV, DAY, PASSON, PASOFF, L, K, LARRV
      COMMON /STNS/ STN, BSTN, N, LSTN, MSTN, LINN, BINN, MINN, B, A, M
      COMMON /HOURS/ IH1,IH2,IH3,IM1,IM2,IM3,IS1,IS2,IS3,LIMITA,IX,IB,D
C
      INTEGER IARRV(300), ISCH(300),IDEP(300),ISCHDV(300),IMINN(50)
      INTEGER IONS(300), IOFFS(300), LOAD(300), ISTN(300), IN, INSTN(50)
      INTEGER PASSON(100), PASOFF(100), M,D
      INTEGER MARR(50), MSTN(50), DAY(10), L, K, STN
      INTEGER LINN, LSTN, N(300), LARRV, MINN(50), B, A
      INTEGER RTENO(10), BLKNO, MO, DATE, YEAR, BARRV, BINN, BSTN
      INTEGER IH1,IH2,IH3,IM1,IM2,IM3,IS1,IS2,IS3,LIMITA,IX,IB
      CHARACTER*30 NAMES(250),NAM1(50),NNAME,NNAM1(50)
C
            N(IB) = D
            M = 1
              DO 66 L = N(IB-1)+1, D
                   IF (IARRV(L).NE.0) THEN
                      MARR(M) = IARRV(L)
                      MINN(M) = LOAD(L)
                      MSTN(M) = ISTN(L)
                      NAM1(M) = NAMES(L)
                    M = M + 1
                   ENDIF
 66           CONTINUE
C
```

```
                DO 77 K =  1, M-2
                   DO 88 J = 1, M-2
                        IF (MINN(J+1).GT.MINN(J)) THEN
                          LINN = MINN(J+1)
                          MINN(J+1) = MINN(J)
                          MINN(J) = LINN
C
                           LSTN = MSTN(J+1)
                           MSTN(J+1) = MSTN(J)
                           MSTN(J) = LSTN
C
                          LARRV = MARR(J+1)
                          MARR(J+1) = MARR(J)
                          MARR(J) = LARRV
C
                          NNAME = NAM1(J+1)
                          NAM1(J+1) = NAM1(J)
                          NAM1(J) = NNAME

                        ENDIF
88                 CONTINUE
77              CONTINUE
C
        IF (MINN(1) .NE. 0) THEN
         IN = IN + 1
          IMINN(IN) = MINN(1)
          INSTN(IN) = MSTN(1)
          NNAM1(IN) = NAM1(1)
        ENDIF
C
        DO 33 A = 1, M-1
           MARR(A) = 0
           MINN(A) = 0
           MSTN(A) = 0
33      CONTINUE
C
             IB = IB + 1
      RETURN
      END
```

ROANOKE VALLEY-METRO APC SYSTEM

TIME POINT-TRIP FILE

ROUTE	BLOCK	DATE
1	17	12/19/85

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	BRENDLES (Time Point)	13:31:30	13:30: 0	- 0: 1:30	5	2	3
34	ROANOKE MEMORIAL HOSPITAL		13:45: 0		10	3	10
49	JEFFERSON & CHURCH	13:56:30	13:55: 0	- 0: 1:30	14	8	9
67	ORANGE & 10 th		14: 3: 0		6	1	14
106	FERNCLIFF APARTMENTS N.		14:15: 0		7	3	13
144	ORANGE & 10 th		14:27: 0		15	7	17
166	JEFFERSON & CHURCH	14:41: 0	14:40: 0	- 0: 1: 0	21	8	22
181	ROANOKE MEMORIAL HOSPITAL		14:47: 0		0	1	21

STOP NO. = 1	MAX. LOAD = 3
STOP NO. = 32	MAX. LOAD = 10
STOP NO. = 43	MAX. LOAD = 11
STOP NO. = 51	MAX. LOAD = 15
STOP NO. = 78	MAX. LOAD = 14
STOP NO. = 111	MAX. LOAD = 17
STOP NO. = 166	MAX. LOAD = 22
STOP NO. = 170	MAX. LOAD = 21
STOP NO. = 209	MAX. LOAD = 22

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	BRENDLES (Time Point)	15: 3:15	15: 0: 0	- 0: 3:15	9	8	23
34	ROANOKE MEMORIAL HOSPITAL	15:18: 0	15:15: 0	- 0: 3: 0	14	1	36
49	JEFFERSON & CHURCH	15:25: 0	15:25: 0	0: 0: 0	4	4	36
67	ORANGE & 10 th		15:33: 0		9	1	44
106	FERNCLIFF APARTMENTS N.		15:45: 0		11	4	43
144	ORANGE & 10 th		15:57: 0		17	5	48
166	JEFFERSON & CHURCH		16:10: 0		20	6	50
181	ROANOKE MEMORIAL HOSPITAL		16:17: 0		20	6	50

STOP NO. = 34	MAX. LOAD = 36
STOP NO. = 49	MAX. LOAD = 36
STOP NO. = 55	MAX. LOAD = 44
STOP NO. = 70	MAX. LOAD = 43
STOP NO. = 142	MAX. LOAD = 48
STOP NO. = 159	MAX. LOAD = 50
STOP NO. = 184	MAX. LOAD = 51

ROANOKE VALLEY-METRO APC SYSTEM

TIME POINT-TRIP FILE

ROUTE	BLOCK	DATE
1	17	12/24/85

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	BRENDLES (Time Point)	9:27:45	9:25: 0	- 0: 2:45	1	0	1
34	ROANOKE MEMORIAL HOSPITAL		9:40: 0		4	0	5
49	JEFFERSON & CHURCH	9:50:30	9:50: 0	- 0: 0:30	5	2	4
67	ORANGE & 10 th	9:58:30	9:58: 0	- 0: 0:30	7	0	11
106	FERNCLIFF APARTMENTS N.		10:10: 0		2	1	12
144	ORANGE & 10 th		10:22: 0		16	6	21
166	JEFFERSON & CHURCH		10:35: 0		21	11	21
181	ROANOKE MEMORIAL HOSPITAL		10:42: 0		22	11	22

STOP NO. = 1	MAX. LOAD = 1
STOP NO. = 33	MAX. LOAD = 5
STOP NO. = 46	MAX. LOAD = 4
STOP NO. = 67	MAX. LOAD = 11
STOP NO. = 85	MAX. LOAD = 12
STOP NO. = 143	MAX. LOAD = 21
STOP NO. = 156	MAX. LOAD = 22
STOP NO. = 170	MAX. LOAD = 22
STOP NO. = 204	MAX. LOAD = 23

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	BRENDLES (Time Point)	10:59:15	10:55: 0	- 0: 4:15	25	24	12
34	ROANOKE MEMORIAL HOSPITAL	11: 9:45	11:10: 0	0: 0:15	3	0	15
49	JEFFERSON & CHURCH	11:22: 0	11:20: 0	- 0: 2: 0	2	4	13
67	ORANGE & 10 th		11:28: 0		7	0	20
106	FERNCLIFF APARTMENTS N.	11:44:45	11:40: 0	- 0: 4:45	8	3	18
144	ORANGE & 10 th		11:57: 0		7	4	21
166	JEFFERSON & CHURCH	12: 8:30	12:10: 0	0: 1:30	20	8	30
181	ROANOKE MEMORIAL HOSPITAL		12:17: 0		3	0	33

STOP NO. = 34	MAX. LOAD = 15
STOP NO. = 49	MAX. LOAD = 13
STOP NO. = 58	MAX. LOAD = 20
STOP NO. = 70	MAX. LOAD = 19
STOP NO. = 112	MAX. LOAD = 23
STOP NO. = 166	MAX. LOAD = 30
STOP NO. = 170	MAX. LOAD = 33
STOP NO. = 183	MAX. LOAD = 34

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD

1	BRENDLES (Time Point)	12:31: 0	12:30: 0	- 0: 1: 0	13	10	33
34	ROANOKE MEMORIAL HOSPITAL	12:45: 0	12:45: 0	0: 0: 0	6	1	38
49	JEFFERSON & CHURCH	12:53:45	12:55: 0	0: 1:15	3	6	35
67	ORANGE & 10 th	13: 3:15	13: 3: 0	- 0: 0:15	8	0	43
106	FERNCLIFF APARTMENTS N.	13:15:30	13:15: 0	- 0: 0:30	2	8	37
144	ORANGE & 10 th		13:27: 0		8	2	43
166	JEFFERSON & CHURCH	13:39:15	13:40: 0	0: 0:45	19	6	50
181	ROANOKE MEMORIAL HOSPITAL	13:45:30	13:47: 0	0: 1:30	2	2	50

STOP NO. = 34 MAX. LOAD = 38
STOP NO. = 42 MAX. LOAD = 37
STOP NO. = 67 MAX. LOAD = 43
STOP NO. = 72 MAX. LOAD = 43
STOP NO. = 143 MAX. LOAD = 43
STOP NO. = 166 MAX. LOAD = 50
STOP NO. = 167 MAX. LOAD = 52
STOP NO. = 184 MAX. LOAD = 50

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	BRENDLES (Time Point)	14: 1:15	14: 0: 0	- 0: 1:15	10	10	50
34	ROANOKE MEMORIAL HOSPITAL		14:15: 0		5	3	52
49	JEFFERSON & CHURCH	14:22:45	14:25: 0	0: 2:15	7	9	48
67	ORANGE & 10 th	14:31:15	14:33: 0	0: 1:45	8	1	55
106	FERNCLIFF APARTMENTS N.		14:45: 0		1	5	51
144	ORANGE & 10 th		15: 2: 0		7	6	56
166	JEFFERSON & CHURCH	15: 9:30	15:15: 0	0: 5:30	9	10	54
181	ROANOKE MEMORIAL HOSPITAL		15:22: 0		8	1	61

STOP NO. = 33 MAX. LOAD = 52
STOP NO. = 45 MAX. LOAD = 49
STOP NO. = 67 MAX. LOAD = 55
STOP NO. = 89 MAX. LOAD = 53
STOP NO. = 112 MAX. LOAD = 56
STOP NO. = 146 MAX. LOAD = 57
STOP NO. = 169 MAX. LOAD = 62
STOP NO. = 209 MAX. LOAD = 61

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	BRENDLES (Time Point)	15:42:30	15:40: 0	- 0: 2:30	23	5	72
34	ROANOKE MEMORIAL HOSPITAL	15:56:45	15:55: 0	- 0: 1:45	7	0	79
49	JEFFERSON & CHURCH	16: 3:30	16: 5: 0	0: 1:30	3	8	74
67	ORANGE & 10 th		16:13: 0		4	1	77
106	FERNCLIFF APARTMENTS N.		16:25: 0		6	9	71
144	ORANGE & 10 th		16:37: 0		9	12	71
166	JEFFERSON & CHURCH		16:50: 0		15	15	74
181	ROANOKE MEMORIAL HOSPITAL		16:57: 0		15	15	74

STOP NO. = 34 MAX. LOAD = 79
STOP NO. = 43 MAX. LOAD = 80
STOP NO. = 50 MAX. LOAD = 77
STOP NO. = 68 MAX. LOAD = 78

```
STOP NO. = 109        MAX. LOAD =  72
STOP NO. = 154        MAX. LOAD =  74
STOP NO. = 203        MAX. LOAD =  73
```

ROANOKE VALLEY-METRO APC SYSTEM

TIME POINT-TRIP FILE

ROUTE	BLOCK	DATE
1	17	1/ 3/86

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	JEFFERSON & CHURCH	10: 7: 0	10: 5: 0	- 0: 2: 0	19	1	18
16	ROANOKE MEMORIAL HOSPITAL		10:12: 0		2	1	19
45	BRENDLES	10:32:30	10:25: 0	- 0: 7:30	5	17	6
78	ROANOKE MEMORIAL HOSPITAL		10:40: 0		1	0	7
93	JEFFERSON & CHURCH	10:49: 0	10:50: 0	0: 1: 0	4	2	8
111	ORANGE & 10 th		10:58: 0		4	1	11
150	FERNCLIFF APARTMENTS N.		11:10: 0		4	2	10
188	ORANGE & 10 th	11:24:45	11:22: 0	- 0: 2:45	15	3	20

STOP NO. =	1	MAX. LOAD =	18
STOP NO. =	2	MAX. LOAD =	19
STOP NO. =	20	MAX. LOAD =	18
STOP NO. =	76	MAX. LOAD =	7
STOP NO. =	93	MAX. LOAD =	8
STOP NO. =	95	MAX. LOAD =	12
STOP NO. =	126	MAX. LOAD =	10
STOP NO. =	188	MAX. LOAD =	20
STOP NO. =	199	MAX. LOAD =	22

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	JEFFERSON & CHURCH	11:35:45	11:35: 0	- 0: 0:45	15	3	32
16	ROANOKE MEMORIAL HOSPITAL	11:42:30	11:42: 0	- 0: 0:30	1	4	29
45	BRENDLES	12: 1:15	11:55: 0	- 0: 6:15	6	12	23
78	ROANOKE MEMORIAL HOSPITAL		12:10: 0		1	1	23
93	JEFFERSON & CHURCH	12:20: 0	12:20: 0	0: 0: 0	4	4	23
111	ORANGE & 10 th	12:30:45	12:28: 0	- 0: 2:45	9	2	30
150	FERNCLIFF APARTMENTS N.		12:45: 0		1	7	24
188	ORANGE & 10 th		12:57: 0		7	9	28

STOP NO. =	2	MAX. LOAD =	33
STOP NO. =	29	MAX. LOAD =	30
STOP NO. =	57	MAX. LOAD =	23
STOP NO. =	93	MAX. LOAD =	23
STOP NO. =	95	MAX. LOAD =	32
STOP NO. =	121	MAX. LOAD =	30
STOP NO. =	185	MAX. LOAD =	28
STOP NO. =	208	MAX. LOAD =	29

STOP #		ARRIVE	SCHED	SCHDEV	PASS. ONS	BET. TIME PTS. OFFS	LOAD
1	JEFFERSON & CHURCH	13: 9:45	13:10: 0	0: 0:15	14	10	34
16	ROANOKE MEMORIAL HOSPITAL	13:16: 0	13:17: 0	0: 1: 0	3	0	37
45	BRENDLES	13:32:30	13:30: 0	- 0: 2:30	7	5	39

78	ROANOKE MEMORIAL HOSPITAL	13:46: 0	13:45: 0	- 0: 1: 0	5	4	40
93	JEFFERSON & CHURCH	13:55:15	13:55: 0	- 0: 0:15	3	4	39
111	ORANGE & 10 th		14: 3: 0		10	2	47
150	FERNCLIFF APARTMENTS N.	14:18:15	14:15: 0	- 0: 3:15	12	6	45
188	ORANGE & 10 th		14:27: 0		10	3	52

STOP NO. = 16 MAX. LOAD = 37
STOP NO. = 44 MAX. LOAD = 39
STOP NO. = 67 MAX. LOAD = 40
STOP NO. = 86 MAX. LOAD = 40
STOP NO. = 102 MAX. LOAD = 47
STOP NO. = 118 MAX. LOAD = 46
STOP NO. = 177 MAX. LOAD = 54
STOP NO. = 195 MAX. LOAD = 53

APPENDIX J. PLOTS PROGRAM

```
C  ************************************************************************
C     PROGRAM NAME : PLOTS  PROGRAM
C  ************************************************************************
C
C     Written By : UMESH D. AVADHANI
C     Date       : NOVEMBER 9, 1985
C
C     Purpose    : READS THE OUTPUT FROM THE FIRST STAGE OF THIS
C                  PROJECT AND THEN AFTER CERTAIN SUITABLE
C                  CALCULATIONS GIVES OUT DATA TO CREATE THE PLOTS.
C
C     UNIT 5 = '  ' (the input to create the ROUTE DEMAND PLOT)
C     UNIT 6 = '  ' (again the input to create the PASS. LOAD PLOT)
C     UNIT 7 = '  ' (this is the output file used as an input to create
C                    the three plots on SYMPHONY )
C
      INTEGER ARRV(300), DAY(31), LOAD(300), RTENO(10), ILOAD(300)
      INTEGER MO, DATE, BLKNO, LIMITA, PASSON(300), ONS(300), YEAR
      INTEGER SSTN, S, P, M, N, B, LaRRV(300), LLOAD(300), LON(300)
      INTEGER STN(50), PASOFF(50), ILOD(50)
      INTEGER CHK(10), LOD(50), W
      REAL ARRL(400), ARR(400)
      CHARACTER*60 NAMES(50)
C
      IA = 0
      RTENO(0) = 1
C
1002  OPEN (1, FILE = ' ')
C
3000  READ (1,9, END = 111) RTENO(B), BLKNO, MO, DATE, YEAR
9     FORMAT(3X, I2, 5X, I3, 4X, I2, 4X, I2, 2X, I2)
C
3001  IF (RTENO(B) .NE. RTENO(B-1)) THEN
      WRITE(*,*) 'START PROCESSING FOR THIS ROUTE ? YES = 1, NO = 0 '
      READ (*,*) N
C
       IF (N .EQ. 0) GO TO 99
        GO TO 2000
      ELSE
        GO TO 1000
      ENDIF
C
2000  IF (RTENO(B) .EQ. 1) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * ' COVE ROAD/PROSPECT HILLS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
```

```
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 2) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'LINCOLN TERRACE/SALEM-SHENANDOAH ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 3) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'RIVERDALE/RUGBY ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 4) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'KENWOOD/VINTON-WISE AVENUE ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 5) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'MELROSE/VIRGINIA HEIGHTS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 6) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'HUNTINGTON COURT/WASENA ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 7) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'RALEIGH COURT/VILLA HEIGHTS ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
```

```
      IF (RTENO(B) .EQ. 8) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'COLONIAL HEIGHTS/WILLIAMSON ROAD ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 9) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'GRANDIN COURT ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF
C
      IF (RTENO(B) .EQ. 10) THEN
        WRITE(*,*)'PLEASE ENTER THE NUMBER OF STOPS IN THE',
     * 'EAST ORANGE AVENUE ROUTE'
        READ(*,*) LIMITA
      B = B + 1
      GO TO 1000
      ENDIF

1000  DO 100 I = 1, LIMITA + 1
C
      IF (I .EQ. LIMITA) GO TO 100
      IF (I .EQ. LIMITA + 1) THEN
       READ (1,*) STN
       GO TO 3000
      ENDIF
C
      READ(1,10,END = 111) ARRV(I), ONS(I), ILOAD(I)
10    FORMAT (35X, I6, 8X, I4, 4X, I4)
C
      IF (ARRV(I) .GE. 21160 .AND. ARRV(I) .LT. 25200) THEN
       J = 1
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 25200 .AND. ARRV(I) .LT. 28800) THEN
       J = 2
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 28800 .AND. ARRV(I) .LT. 32400) THEN
       J = 3
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
```

```
      ENDIF
C
      IF (ARRV(I) .GE. 32400 .AND. ARRV(I) .LT. 36000) THEN
       J = 4
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 36000 .AND. ARRV(I) .LT. 39600) THEN
       J = 5
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 39600 .AND. ARRV(I) .LT. 43200) THEN
       J = 6
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 43200 .AND. ARRV(I) .LT. 46800) THEN
       J = 7
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 46800 .AND. ARRV(I) .LT. 50400) THEN
       J = 8
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 50400 .AND. ARRV(I) .LT. 54000) THEN
       J = 9
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 54000 .AND. ARRV(I) .LT. 57600) THEN
       J = 10
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 57600 .AND. ARRV(I) .LT. 61200) THEN
       J = 11
          PASSON(J) = ONS(I) + PASSON(J)
          GO TO 100
      ENDIF
C
      IF (ARRV(I) .GE. 61200 .AND. ARRV(I) .LT. 64800) THEN
```

```
        J = 12
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
        IF (ARRV(I) .GE. 64800 .AND. ARRV(I) .LT. 68400) THEN
         J = 13
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
        IF (ARRV(I) .GE. 68400 .AND. ARRV(I) .LT. 72000) THEN
         J = 14
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
        IF (ARRV(I) .GE. 72000 .AND. ARRV(I) .LT. 75600) THEN
         J = 15
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
        IF (ARRV(I) .GE. 75600 .AND. ARRV(I) .LT. 79200) THEN
         J = 16
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
        IF (ARRV(I) .GE. 79200 .AND. ARRV(I) .LT. 82800) THEN
         J = 17
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
        IF (ARRV(I) .GE. 82800 .AND. ARRV(I) .LT. 84600) THEN
         J = 18
           PASSON(J) = ONS(I) + PASSON(J)
           GO TO 100
        ENDIF
C
100     CONTINUE
C
 111     DO 500 IL = 1, J
           LON(IL) = PASSON(IL)
 500     CONTINUE
C
        WRITE (*,*) 'PROCESS DATA FOR LOAD PROFILE PLOT ?, YES = 1,NO =0'
        READ (*,*) P
        IF (P .EQ. 0) GO TO 99
        CLOSE(1, STATUS = 'KEEP')
```

```
      OPEN (4,FILE = ' ')
      WRITE (*,*) LIMITA
 200   DO 400 K = 1, LIMITA + 1
      IF (K .EQ. 1) GO TO 400
      IF (K .EQ. LIMITA) GO TO 400
      IF (K .EQ. LIMITA + 1) THEN
         READ(4,*) SSTN
         GO TO 200
      ENDIF
C
      READ(4,11,END = 600) LARRV(K), LLOAD(K)
 11    FORMAT (35X, I6, 16X, I4)
      IF (LARRV(K) .NE. 0) THEN
       IB = IB + 1
          LOAD(IB) = LLOAD(K)
          ARRL(IB) = LARRV(K)/3600.00
      ENDIF
C
400   CONTINUE
C
600   DO 503 IU = 1, IB
      IF (ARRL(IU) .LT. 1 .OR. ARRL(IU) .LE. 0) GO TO 503
       II = II + 1
       ARR(II) = ARRL(IU)
503   CONTINUE
C
      OPEN (UNIT = 5, FILE = ' ')
C
      CHK(1) = 1
      CHK(2) = 34
      CHK(3) = 49
      CHK(4) = 67
      CHK(5) = 106
      CHK(6) = 144
      CHK(7) = 166
      CHK(8) = 1181
C
8000  DO 700 W = 1, 8
      READ (5,14,END=601) STN(W), NAMES(W),LOD(W)
14    FORMAT (2X, I3, 3X, A27, 14X, I4)
C
      IF (STN(W) .EQ. CHK(1)) ILOD(1) = LOD(W) + ILOD(1)
      IF (STN(W) .EQ. CHK(2)) ILOD(2) = LOD(W) + ILOD(2)
      IF (STN(W) .EQ. CHK(3)) ILOD(3) = LOD(W) + ILOD(3)
      IF (STN(W) .EQ. CHK(4)) ILOD(4) = LOD(W) + ILOD(4)
      IF (STN(W) .EQ. CHK(5)) ILOD(5) = LOD(W) + ILOD(5)
      IF (STN(W) .EQ. CHK(6)) ILOD(6) = LOD(W) + ILOD(6)
      IF (STN(W) .EQ. CHK(7)) ILOD(7) = LOD(W) + ILOD(7)
      IF (STN(W) .EQ. CHK(8)) ILOD(8) = LOD(W) + ILOD(8)
      IF (STN(W) .EQ. CHK(8)) GO TO 8000
700   CONTINUE
```

```
C
601     DO 501 IL = J+1, II
           LON(IL) = 0
           STN(IL) = 0
          ILOD(IL) = 0
 501    CONTINUE
C
       WRITE (*,*) 'ANY OTHER DATA FILE TO BE PROCESSED?, YES = 1, NO = 0
     *'
       READ (*,*) MM
       IF (MM .EQ. 1) GO TO 1002
C
       OPEN (2, FILE = ' ', STATUS = 'NEW')
C
       WRITE (2,25)
       WRITE (2,20)
C
       DO 300 L = 1, IL - 1
        IF (ILOD(L) .NE. 0 .OR. STN(L) .NE. 0) THEN
         WRITE (2,17) ARR(L), LOAD(L), LON(L), STN(L), NAMES(L), ILOD(L)
         WRITE (*,*) ARR(L), LOAD(L), LON(L), STN(L), NAMES(L), ILOD(L)
        ELSEIF (LON(L) .NE. 0) THEN
          WRITE (2,12) ARR(L), LOAD(L), LON(L)
          WRITE (*,*) ARR(L), LOAD(L), LON(L)
        ELSE
          WRITE(2,13) ARR(L), LOAD(L)
          WRITE(*,*) ARR(L), LOAD(L)
        ENDIF
 300    CONTINUE
 12     FORMAT(2X, F5.2, 5X, I3, 7X, I3)
 13     FORMAT(2X, F5.2, 5X, I3)
 17     FORMAT(2X, F5.2, 5X, I3, 7X, I3, 11X, I3, 4X,'"',A26,'"',1X, I3)
 25     FORMAT (4X,'"COVE ROAD/PROSPECTS HILLS ROUTE"',/)
 20     FORMAT ('"ARRIVAL"', 2X, '"LOAD"', 2X, '"PASS. ONS"', 2X, '"TIME
     *PT"', 8X, '"NAME"', 13X, '"LOAD"',/)
99      END
```

The vita has been removed from the scanned document