Weighing Vehicles in Motion [1969]

University of Kentucky Research Foundation
MEMORANDUM TO: A. O. Neiser
State Highway Engineer
Chairman, Research Committee

SUBJECT: Research Report (Interim); "Weighing Vehicles in Motion," KYHPR-61-27; HPR-1 (5), Part II

The report forwarded herewith covers the final phase of research and development contracted by the University of Kentucky Research Foundation in behalf of the subject study; this report, together with an operational installation and data analysis system, fulfills all objectives and commitments assigned.

The earlier phase of this work, which began in 1961, was devoted to the design of a suitable weighing platform to be installed in the pavement, in a traffic lane. Of several designs investigated, a "broken-back" platform -- that is, two simple spans with abutting ends supported commonly on load-cells -- was judiciously selected. This type of platform produces a triangular output signal from the load cells as a load (axle) traverses the platform. The apex or peak of the triangular signal from the load cells is calibrated in weight units. The unique feature of this type of design is that the base leg of the triangle represents the span length; the addition of an internal timing signal permits speed-of-traverse to be calculated; then, by presetting a practical time-gap between vehicles, it is possible to determine the number of axles per vehicle (classification) and to sum the several axleloads -- yielding a gross load for each vehicle. Thus, the digitized output capabilities of the system are: 1) load impulses of individual axles, 2) vehicle speed, 3) gross load, and 4) vehicle classification -- that is, by number of axles. Various statistical analyses may be programmed to determine specific characteristics of the traffic stream.

The axleloads sensed by this system are most necessarily equivalent to static weights. Vehicles in motion tend to undulate or bounce as they travel; there is a random probability or likelihood that a vehicle (or axle) will be on an "upsing" or "downswing" when it crosses the platform. The most unlikely events would be to catch an axle at either extreme or at its null (equivalent static) state; however, there is a greater probability that an axle will be closer to a null condition than to an extreme as it crosses the platform. The standard error of estimate is adjudged to be in the order of ± 5% of the static weight (*cf. previous report, April 1964). Statistically speaking, the errors tend to cancel; and so the use of the system for survey purposes is not impaired.

Whereas, the scale system is capable of measuring the force exercised by a set of wheels moving at high speeds, the force impressed on the platform is simply not the static weight force of the axle. The ratio of the peak downward forces to the static weight force defines "Impact Factor". This explanation merely emphasizes the fact the weighing platform senses only the instantaneous, dynamic force of each transient axle. If it were possible, by some contrivance, to measure the static weight force instead of the instantaneous dynamic force, some knowledge of impact factors would be needed in order to translate back to the dynamic state.
The Research Division planned to operate the installation on a pilot basis for at least one year following completion of the Research Foundation's proof-testing of the completed system. We were not privileged to proceed until the Bureau of Public Roads had reviewed a draft of this report and authorized continuation (cf. KYHPR-61-27; Part II, HPR-1 (5)). Approval of the continuation plan was received in December. Reports covering the pilot period will be forthcoming.

The collection of traffic-load statistics has traditionally been considered to be a planning function. ATR's together with out-of-stream weighing scales (static) presently provide the principal data. As originally conceived, an in-stream system could provide more meaningful data. The use of the system (Implementation) remains optional with the Planning Division. Hopefully, the results obtained from the pilot-period of operation will prove to be enabling and persuasive.

The system has been dubbed "RODAS" (Road Data Acquisition System). The electronic digitizing system may have applications other than in couple with a weighing platform implanted in a pavement. With minor alterations, the system might be adopted to monitor any SR 4-type strain transducer. Some consideration has been given to the possibility of implanting a weighing platform into major bridges and to the further possibility of monitoring attendant strains in critical structural members -- in order to develop fatigue histories (cf. KYP-69-11; Part III-A, HPR-1 (5)).

Respectfully submitted

Jas. H. Havens
Director of Research

JHH:slw
Enclosure
cc's: Research Committee
State Highway Engineer, Chairman
All Assistant State Highway Engineers
Assistant Pre-Construction Engineer
Assistant Operations Engineer
Executive Director, Office of Computer Services
Executive Director, Office of Equipment & Properties
Director, Division of Bridges
Director, Division of Construction
Director, Division of Design
Director, Division of Maintenance
Director, Division of Materials
Director, Division of Photogrammetry
Director, Division of Planning
Director, Division of Research
Director, Division of Right of Way
Director, Division of Roadside Development
Director, Division of Rural Roads

Director, Division of Traffic
All District Engineers
Division Engineer, Bureau of Public Roads
Chairman, Department of Civil Engineering, U. K.
Associate Dean for Continuing Education, U. K.
This report was prepared by the Office of Research and Engineering Services, College of Engineering, in cooperation with the University of Kentucky Research Foundation, covering Contract No. CH-13043 with the Kentucky Department of Highways (UKRF Project No. 2C1-65-00709-83001), for the period March 15, 1966 through June 30, 1969. Principal investigators for this research were David K. Blythe, Chairman, Department of Civil Engineering, and Russell E. Puckett, Associate Director, Office of Research and Engineering Services.

UNIVERSITY OF KENTUCKY RESEARCH FOUNDATION

in cooperation with the
Division of Research
DEPARTMENT OF HIGHWAYS
Commonwealth of Kentucky

and

U.S. Department of Transportation
Federal Highway Administration
Bureau of Public Roads

The opinions, findings, and conclusions in this report are not necessarily those of the Department of Highways or the Bureau of Public Roads

November 1969
RESEARCH RELATING TO WEIGHING VEHICLES IN MOTION

INTRODUCTION

An agreement between the Kentucky Department of Highways and the University of Kentucky Research Foundation, dated March 15, 1969, extended and continued research on Weighing Vehicles in Motion. The agreement contained these objectives of the continuing research:

"The services provided by the FOUNDATION shall be within the scope of the objectives outlined in the original proposal and made a part of the original contract (Research Project KYHPR-61-27, Development of an Electronic Means of Weighing Vehicles in Motion, a part of the DEPARTMENT'S Work Program, HPR-1, Part II). The research shall consist of, but not limited to, the following:

A. Furnishing technical assistance as needed to effectuate the installation of the scale platform and its accessories at the chosen site and in accordance with plans contained in the FOUNDATION’S annual report of “Weighing Vehicles in Motion,” Submitted to the DEPARTMENT, September 30, 1965. Any revisions or modifications arising shall be subject to the concurrence of the DEPARTMENT. The research staff shall record all pertinent details of the installation and submit a descriptive report subsequently.

B. Procuring, assembling, testing, and installing the recording system recommended in the annual report (op. cit.) and in attachment No. 1, “Proposed Budget,” which is appended hereto.

C. Developing and testing a data-analysis program satisfying the modular input requirements of available digital computers and the desired format of "output".

Changes in the area or topics designated for study were permitted, based on consultation and approval by the parties to the agreement. In addition, the Department reserved the right to revise the scope of the research.

The Department of Civil Engineering, University of Kentucky, was chosen to conduct the research under the direction of David K. Blythe. His staff designed, constructed, and installed the broken-bridge scale platform at the preselected site on I-64-75, and purchased an electronic data-collection system for installation at the site. Since July 1, 1968, the project has continued under the direction of Russell E. Puckett, Associate Director, Office of Research and Engineering Services, College of Engineering.

In accordance with the provisions of the agreement, the data-collection system was selected for its ability to detect individual axles of moving vehicles and to record data in digital format on magnetic tape. The system was designed to convert both the analog signal representing axle weight and a signal representing real time to a digital form that would be compatible with available digital computer equipment at the University of Kentucky.

The separate objectives of the research have been completed. This report includes sections devoted to each of the objectives, and is submitted for approval and acceptance as the final report of this contract agreement.

INSTALLATION OF THE SCALE PLATFORM

The scale platform and frames were assembled during the preceding project period. Various views of the completed scale are shown in Figs. 1-5.
Fig. 1. Components of the broken-bridge scale include the frame, two halves of the platform, and spring preloading members.

Fig. 2. Close-up view of the frame of the broken-bridge scale.
Fig. 3. Preload members are shown in front of the scale platforms.

Fig. 4. Cover plates that are installed on the platform halves and set flush with the highway pavement.
Fig. 5. View of a load cell and its leveling mechanism installed in the scale.
A pit for the scale was constructed in the outside, eastbound lane of Interstate Highways 64 and 75, just east of the Bryan Station Road overpass near Lexington, Kentucky. Details of the site are shown in Fig. 6, and the location of the scale is shown more clearly in Figs. 7 and 8. This site was selected because of good sight distances on a long, nearly level tangent, and because it permitted the instrument house and operating personnel to be shielded from the view of approaching traffic (behind the abutment fill for the overpass bridge). The completed scale was installed in the pit on July 19, 1966, and opened for traffic on July 22, 1966. (Plan drawings and details for the broken-bridge scale platform and pit construction were included in the Annual Report on KYHPR-61-27, dated September 1965.)

**Maintenance Problems**

Several problems were experienced with the scale installation during the current project period. After about six months of traffic over the platform, the allenhead stud bolts that secure the skid plate deck of the scale to the platform frame and to the outside frame worked loose, and several bolts were broken at the junction of their cone-shaped head and the shank. After the first occurrence of this maintenance problem, the bolts were replaced and tightened securely, using a special adhesive with each bolt.

During the winter of 1966, several bolts were broken and a small piece at one corner of the platform was sheared off by a snowplow. Further repairs of the scale included the addition of small, steel blocks placed between the tops of the transverse safety beams and the bottoms of the platform frames, at both the leading and trailing edges. These blocks were designed to provide fixed, rocker-type supports at the leading and trailing edges of the platform, in an attempt to reduce the shear loads on the stud bolts. As a further refinement of the system, the outside half-plates on each side of the platform were welded to the outside frame. The inside half-plates covering the scale remained bolted, to provide a means of access to the load cells and the leveling mechanisms.

During the April-June 1968 quarter, the scale platform had to be repaired. Broken bolts were replaced, and it was noted that the welds on the trailing edge of the scale were broken at the outside corner of the platform. No mechanical malfunction occurred, but the cover plate was loosened by the broken bolts. The platform and load cells continued to operate after thousands of load applications at the testing site, in spite of these minor maintenance problems.

**Defective Load Cell**

The platform scale operated in a satisfactory manner since its initial installation, except for the problems with broken and loosened bolts, until February 1965. In making preparations to install the electronic data-collection system at the site, it was found that one of the two load cells was inoperative. Since both cells had been used in the project for several years, including the testing program at the I-64 site near Shelbyville, Kentucky, it was decided to replace both cells. It was also decided to recondition the cover plates on the platform while the load cells were being replaced, in an attempt to preclude the problems previously encountered with the cover plates and bolts.

The defective load cell was disassembled in an attempt to determine the cause of failure. Because of its construction, it was difficult to dismantle the cell so that electrical connections were undisturbed. It was found that the gages were still bonded to the load-bearing column and remained connected in their bridge configuration. A broken wire was found between the bridge and the cable connection, but its cause could not be determined. No other cause of failure could be found, so it was assumed that the broken wire had made the cell inoperative.

**Modification of Cover Plates**

While the load cells were being replaced, the cover plates were modified before being reinstalled, as shown in Fig. 9. The small, cutout locations were chosen to allow access to the leveling and preloading
**Fig. 6.** The scale platform is installed just to the rear of the automobile in the outside lane. The instrument house is shown at the left.
Fig. 7. The Instrument house and scale platform as seen from the bridge overhead.
Fig. 8. The truck is approaching the leading edge of the scale.
mechanisms from the top of the scale. The original design of the scale necessitated removal of the entire platform assembly in order to gain access to the load cells for removal and replacement. Because the scale is in the traffic lane, it was necessary to divert the traffic at the site to a single lane for a period of four days. These conditions during maintenance of the scale introduced safety hazards to the personnel, and indicated that future installations should provide access to the load cells and the underground structure through a manhole and a deeper pit to contain the scale platform.

ELECTRONIC DATA-COLLECTION SYSTEM

The purchase order for the electronic data system was issued on February 1, 1967, to Robert Perelman, dba DGE Instruments, University Heights, Ohio. Specifications of the system were as follows:

1. The vehicle may have from 2 to 10 axles. Current plans are based on 5 with provision for expansion to 7.

2. The vehicle may travel at legal speeds up to 70 miles per hour; therefore, an actual timing of up to 80 would be desirable.

3. Dynamic loads will range from 1,000 to 30,000 pounds per axle.

4. The dynamic measurement load accuracy should be recorded with an accuracy of plus or minus 200 pounds per axle.

5. The speed of the vehicle must be recorded or deduced at an accuracy of plus or minus 5 miles per hour.

6. The system will be designed so that the number of axles per vehicle can be deduced from the data on the digital tape.

7. It is desirable to be able to deduce the spacing between axles within plus or minus 1/2 foot.

8. It is desirable that the measurement system operate unattended over a minimum period of 24 hours, with a longer period preferred.

9. Dual-axle trucks may have load of two axles on the scale simultaneously. Therefore, under this condition, the trace will not return to zero between axles. In addition, the noise pips on the sides of the signal are characteristic and must not be accepted as peaks of minor waves.

10. It should be possible to determine the approximate time of day and date for the passage of each vehicle, from the data on the digital tape.

The Signal Conditioner

The project staff started the design of a signal conditioner, to be used as the interface between the load cells and the digital data system, as reported in the June 1967 monthly report. This type of interface was necessary because the vendor had been furnished an ideal, triangular signal waveform from which to design the data system. However, the load cell signal that corresponds to applied weight approximates the ideal signal only as an "envelope" about its peaks. In addition, the scale platform and moving vehicle introduce spurious noise signals caused by their vibrations that were referred to in specification (9) above. A typical waveform is shown in Fig. 10.

In order to generate a signal from the load cells that would approach the waveform of the ideal signal, it was necessary to construct a signal conditioner that was capable of producing a nearly triangular
Fig. 9. Changes in cover plate design.
Fig. 10. Oscilloscope waveforms show action of the active filter; upper trace is load-cell signal, lower trace is output from the filter and input to data-collection circuitry; obtained for a 3-axle truck.
waveform as the input signal for the digital data system. This unit had been constructed and installed and was operated satisfactorily before the digital system was delivered and installed on August 21, 1967.

When it was delivered and installed, the digital data system was not capable of distinguishing between high-speed axles, and it responded to the noise signals arising from platform and vehicle oscillations. The vendor was not able to remain at the site long enough to adequately check out his system, and strongly suggested that the signal conditioner was causing malfunctions in the data system, even though the data system had been designed to eliminate response to the spurious noise signals.

The project staff then began the design for a filter network that would be effective in reducing or eliminating these noise signals. In February 1968, the design of an active filter was completed. Its operation completely negated the complaints about noisy input signals to the data system, because this active network passed a signal that represented only the weight of an axle on the platform and completely eliminated components due to vehicle and platform oscillations. Its output waveform, Fig. 10, closely approximated the ideal, nearly triangular waveform.

Additional Circuit Designs

In April 1968, the vendor returned to the installation site for further testing of the data system, parts of which had been rebuilt. Many deficiencies and malfunctions remained in the system. At that time, the vendor asked to be relieved of further responsibilities in the design and installation of the electronic data system, because of changes in his employment.

The vendor was relieved from the conditions of the purchase order, and the Office of Research and Engineering Services, College of Engineering, accepted the task of completing the design and installation of the data system, in July 1968. An inspection of the individual circuit cards making up the logic and data pulse handling circuits showed that an extensive redesign was necessary, in order to meet the specifications of the system. Tests showed that the system contained several design errors, including improper logic and data pulse sequences. In addition, the data system could handle only one axle at a time, precluding the necessary storage of data for high-speed tandem axles and multi-axle vehicles. Additional buffer storage was added to the circuitry to permit storage and simultaneous processing of up to four axles. Additional axle-data storage is unnecessary at the present site, because the system can handle existing traffic with buffer storage for up to four axles.

Detailed descriptions and performance data for the electronic data system are included in the Operation and Instruction Manual, Appendix B, and made a part of this report.

DATA-ANALYSIS PROCEDURES

Format of Recorded Field Data

For each axle weighed by the scale and recorded on the digital tape recorder, a sequence of 21 characters is used to include data representing axle weight, speed of the vehicle, date and time the event occurred, and a number assigned to the location of the weighing station. As an example of the data sequence, one axle may generate this sequence of 21 characters: /14700623652359599997. These characters are recorded on the magnetic tape in the given sequence, and represent data in this manner: the "slash" character is used to separate data for successive axles; 147 represents the weight of the axle; 0062 is the number of milliseconds the axle was on the scale, hence, is a measure of speed of the vehicle; 365 is the day of the year; 2359 is hour and minute the axle was weighed; 59999 is the time when the axle was weighed, in seconds to the nearest millisecond; the final 7 is the number assigned to the weighing station where the data were recorded (this number is entered manually on the data-collection system, and can be any digit from 0 to 9). Each axle that is weighed generates a stream of 21 similar characters, starting with the "slash" as the first data byte.
Computer Program for Data Analysis

A data-analysis program was prepared in PL/1 language for use on the University's IBM 360/50 computer, to analyze field tapes containing the raw field data recorded from the scale. The complete program is given in Appendix A. Samples of the output, given for each hour and including calculated maximum, mean, and minimum values of weight, speed, and axle spacing, are shown in Fig. 11. The printed output includes information on the number of vehicles processed. These vehicles for which all information cannot be calculated, either because of extraneous characters on the tape or for other reasons, are identified as "invalid vehicles" and printed. Similarly, the program separates data representing a "test vehicle" and prints information about its weight, speed, and axle spacing. (A test vehicle is identified by the program from the recorded digit "9" in the data byte representing the weighing station number.) These printed values can be used to evaluate the operation of a particular weighing station, to determine whether the overall weighing system is functioning properly, based on the computer analysis of recorded data.

The analysis program contains parameters that are related to specific properties of a given weighing station. For example, speed calculations are based on a value that has been determined as the "effective width" of the scale platform in the direction of vehicular traffic. This value can be determined from measurements of a test vehicle whose axle spacing is known and whose speed can be monitored and controlled. From measurements of the length of time the test vehicle was on the scale platform, together with its speed and its axle spacing, the "effective width" of the scale at the I-64-J-5 site was determined to be 4.313 feet. This value is used in the computer program to calculate speeds and axle spacings.

The program also recognizes an "overload" weight calculation whenever the data bytes representing weight exceed 199. The program enters the digits 800 whenever an overload occurs, and will show up in the printed output as 80.0 kips for the overloaded axle. A factor is included in the program to calculate weight according to a preselected ratio between the data bytes and dynamic weight of the axle, based on the calibration factor that is set into the data-collection circuitry. A typical calibration might correlate the data bytes 200 with 20,000 pounds. The number 200 can be set in the electronic circuitry to represent weight in the range 10,000 to 30,000 pounds. Whatever maximum weight is chosen in this calibration will be interpreted by the computer program as the "overload" value.

Another feature of the computer program identifies and separates vehicles, based on a maximum axle spacing of 35 feet. If the axle spacing is calculated to be greater than 35 feet, the program recognizes the end of one vehicle and starts accumulating data for another vehicle. The program can accumulate data for vehicles having up to 9 axles.

Stored Information

Data for vehicles for which all parameters can be calculated ("valid vehicles"), are stored on a second, intermediate magnetic tape associated with the computer, during the analysis of the field tape. Stored data consist of characters for each axle, included in a sequence of digits to represent various parameters. A typical sequence could be, 18407421473652359599937/. The characters 184 represent the axle weight, calculated to the nearest 100 pounds, e.g., 18.4 kips; 0742 is the speed, calculated to the nearest tenth mph, e.g., 74.2 mph; 147 is the calculated axle spacing, e.g., 14.7 feet; 365 is the day; 2359 is hour and minute when the axle was weighed; 59999 is the time of weighing, in seconds to the nearest millisecond; 3 is the axle number in the sequential weighing of axles on the same vehicle; 7 is the number assigned to the weighing station where data were obtained; and the final "slash" character separates these data from those for a succeeding axle, either the fourth axle of this vehicle or the first axle of the next one. Only valid vehicles are included on this intermediate tape during the analysis of field tapes. All other vehicle data are contained in the printed information generated by the computer, which is not stored but appears only as the printed output during analysis of the field tape.
*************** BEGIN VEHICLE DATA FOR THE HOUR 13 TO 14 ***************
DAY = 358, STATION IDENTIFICATION = 4
NUMBER OF VALID VEHICLES FOR THIS HOUR = 74
INVALID VEHICLES = 4, TEST VEHICLES = 0
NUMBER OF AXLES FOUND THIS HOUR = 205

<table>
<thead>
<tr>
<th>MAXIMUM</th>
<th>MINIMUM</th>
<th>AVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL AXLE SPEED</td>
<td>162.8 MPH</td>
<td>32.2 MPH</td>
</tr>
<tr>
<td>FIRST AXLE SPEED</td>
<td>162.8 MPH</td>
<td>35.3 MPH</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>17.3 KIPS</td>
<td>0.7 KIPS</td>
</tr>
<tr>
<td>AXLE SPACING</td>
<td>32.2 FEET</td>
<td>2.5 FEET</td>
</tr>
</tbody>
</table>

INVALID VEHICLE NO. 1
AXLE: NUMBER  WEIGHT  SPEED  SPACING
1  5.9 KIPS  162.8 MPH  0.0 FEET

INVALID VEHICLE NO. 2
AXLE: NUMBER  WEIGHT  SPEED  SPACING
1  2.8 KIPS  49.6 MPH  0.0 FEET
2  1.1 KIPS  49.6 MPH  10.3 FEET
3  1.4 KIPS  49.6 MPH  0.0 FEET
4  1.6 KIPS  49.6 MPH  0.0 FEET
5  2.6 KIPS  49.6 MPH  7.3 FEET

INVALID BECAUSE OF INABILITY TO DISTINGUISH BETWEEN VEHICLES
AXLE: NUMBER  WEIGHT  SPEED  SPACING  ID
1  1.7 KIPS  73.2 MPH  0.0 FEET  4
2  1.6 KIPS  73.2 MPH  0.0 FEET  4
3  7.7 KIPS  73.2 MPH  0.0 FEET  4
4  14.2 KIPS  73.2 MPH  0.0 FEET  4
5  13.1 KIPS  73.2 MPH  0.0 FEET  4
6  15.1 KIPS  73.2 MPH  12.6 FEET  4
7  17.3 KIPS  73.2 MPH  7.9 FEET  4
8  12.3 KIPS  73.2 MPH  32.2 FEET  4
9  1.4 KIPS  73.2 MPH  7.9 FEET  4
10 2.2 KIPS  73.2 MPH  5.5 FEET  4

Fig. 11. Typical printed output from analysis of field tape.
### INVALID VEHICLE NO. 3

<table>
<thead>
<tr>
<th>AXLE NUMBER</th>
<th>WEIGHT</th>
<th>SPEED</th>
<th>SPACING</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2 KIPS</td>
<td>75.1 MPH</td>
<td>0.0 FEET</td>
</tr>
<tr>
<td>2</td>
<td>1.6 KIPS</td>
<td>75.1 MPH</td>
<td>0.0 FEET</td>
</tr>
<tr>
<td>3</td>
<td>0.9 KIPS</td>
<td>75.1 MPH</td>
<td>0.0 FEET</td>
</tr>
<tr>
<td>4</td>
<td>1.6 KIPS</td>
<td>75.1 MPH</td>
<td>12.2 FEET</td>
</tr>
<tr>
<td>5</td>
<td>2.5 KIPS</td>
<td>75.1 MPH</td>
<td>15.6 FEET</td>
</tr>
</tbody>
</table>

### INVALID VEHICLE NO. 4

<table>
<thead>
<tr>
<th>AXLE NUMBER</th>
<th>WEIGHT</th>
<th>SPEED</th>
<th>SPACING</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.3 KIPS</td>
<td>88.8 MPH</td>
<td>0.0 FEET</td>
</tr>
</tbody>
</table>

### GROSS VEHICLE NO. SPEED WEIGHT AXLE SPACING (FEET)

<table>
<thead>
<tr>
<th>VEHICLE NO.</th>
<th>SPEED</th>
<th>WEIGHT</th>
<th>1-2</th>
<th>2-3</th>
<th>3-4</th>
<th>4-5</th>
<th>5-6</th>
<th>6-7</th>
<th>7-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77.1 MPH</td>
<td>4.0 KIPS</td>
<td>10.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>83.7 MPH</td>
<td>1.5 KIPS</td>
<td>9.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>79.2 MPH</td>
<td>3.7 KIPS</td>
<td>10.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>73.2 MPH</td>
<td>3.0 KIPS</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>68.1 MPH</td>
<td>4.1 KIPS</td>
<td>10.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>49.6 MPH</td>
<td>52.3 KIPS</td>
<td>8.4</td>
<td>4.1</td>
<td>14.1</td>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>69.7 MPH</td>
<td>3.8 KIPS</td>
<td>10.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>81.4 MPH</td>
<td>1.7 KIPS</td>
<td>11.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>45.1 MPH</td>
<td>53.3 KIPS</td>
<td>7.6</td>
<td>2.5</td>
<td>5.4</td>
<td>14.4</td>
<td>3.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>73.2 MPH</td>
<td>2.2 KIPS</td>
<td>11.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>66.6 MPH</td>
<td>4.2 KIPS</td>
<td>11.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>73.2 MPH</td>
<td>3.9 KIPS</td>
<td>9.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>53.3 MPH</td>
<td>24.0 KIPS</td>
<td>8.2</td>
<td>5.1</td>
<td>19.0</td>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>66.6 MPH</td>
<td>4.2 KIPS</td>
<td>10.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>75.1 MPH</td>
<td>3.5 KIPS</td>
<td>9.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>65.1 MPH</td>
<td>4.7 KIPS</td>
<td>11.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>66.6 MPH</td>
<td>4.2 KIPS</td>
<td>14.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>73.2 MPH</td>
<td>3.8 KIPS</td>
<td>11.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>50.8 MPH</td>
<td>3.2 KIPS</td>
<td>15.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>75.1 MPH</td>
<td>3.2 KIPS</td>
<td>10.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>77.1 MPH</td>
<td>2.2 KIPS</td>
<td>11.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>77.1 MPH</td>
<td>3.3 KIPS</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>79.2 MPH</td>
<td>3.9 KIPS</td>
<td>10.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>83.7 MPH</td>
<td>3.9 KIPS</td>
<td>9.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>81.4 MPH</td>
<td>3.0 KIPS</td>
<td>10.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>81.4 MPH</td>
<td>2.7 KIPS</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>79.2 MPH</td>
<td>1.8 KIPS</td>
<td>11.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 11. (Con't) Typical printed output from analysis of field tape.
Computer Program for Intermediate Tape Analysis

Data that are stored on the intermediate tape include all axles of vehicles that have been identified as individual vehicles, based on the test of 35 feet minimum distance between the last axle of one and the first axle of the following vehicle. Some of these data may include axle weights, speed, and axle spacings that were not calculated (entered as zero values), because of scattered, extraneous characters on the field tape. Only those axles that are identified with a vehicle, based on the axle-spacing test, are included on the intermediate tape. All other axles have been culled and printed as “invalid vehicles” during the analysis of the field tape.

The program prepared in PL/1 language for analysis of intermediate tapes is shown in Appendix A. This program generates an output in the form of tabular information relating type of vehicle, axle weights, speeds, and other parameters chosen by the Kentucky Department of Highways. Typical output is included in Figure 12. The category codes for different vehicles can be explained by reference to Figure 13. Categories have been established for up to six axles per vehicle.

CONCLUSIONS

1. The data-collection system that was designed and developed in accordance with the contract agreement provides an automatic means of measuring not only the weights of moving vehicles, but also their speed and axle spacings, as well as date and time of weighing, number and spacing of axles, and headway between successive vehicles that pass over the weighing platform. The techniques used in the system described herein could be expanded to include all traffic lanes, using a separate scale platform for each lane and an expanded buffer storage within the electronic circuitry for storage of additional axle data.

2. Unattended operation of the data-collection system permits collection of desired data at times that may be inconvenient for staff personnel.

3. The electronic circuitry is compatible with other transducers that generate the triangular waveform that is typical of the load cell response in this weighing system.

RECOMMENDATIONS

1. The pit that contains the scale platform should be enlarged and designed to permit access to the load cells and the leveling and preloading mechanisms, from a manhole located outside the traffic lane. Safety hazards to maintenance personnel generated by traffic in contiguous lanes as well as hazards to traffic past the platform after daylight hours, should be considered as primary concerns in the design of other installations.

2. The digital data-collection system should be installed in a mobile vehicle, such as a panel truck, to permit the same system to be used at more than one weighing station. Its mobility would also preclude problems of vandalism at the weighing stations.

3. The prototype system installed at the I-64-75 site should be subjected to an evaluation of its performance and characteristics for a period of at least one year, in order to determine effects of temperature and other weather factors.

4. To allow for necessary design changes, particularly for mobile operation and space and weight reduction, additional installations of weighing stations that will use the data-collection system of this prototype design should be planned so that necessary lead times for receipt of materials and construction of the system can be accommodated.

5. No long-term analysis of collected data has been completed, because it was not a part of the contract agreement. The relationships of dynamic and static weights of moving vehicles, correlations of weather, time of year, day of the week, time of day, etc., with type of vehicle and its load are apparent subjects for extensive study using the data collected by this type of system.
Fig. 12. Typical output data from intermediate tapes.

GROSS OPERATING WEIGHT VERSUS AXLE PLACEMENT (AASHTO CATEGORIES)

TANDEM SPACING IS 40 INCHES OR LESS
1.2 AND 3 INDICATE SINGLE, BITANDEM AND TRITANDEM AXLES

<table>
<thead>
<tr>
<th>AXLE PLACEMENT</th>
<th>UNDER 4 TONS</th>
<th>4.0-6.0</th>
<th>6.0-8.0</th>
<th>8.0-10.0</th>
<th>10.0-12.0</th>
<th>12.0-14.0</th>
<th>14.0-16.0</th>
<th>16.0-18.0</th>
<th>18.0-20.0</th>
<th>20.0-24.0</th>
<th>24.0-28.0</th>
<th>28.0-32.0</th>
<th>32.0-36.0</th>
<th>36.0-40.0</th>
<th>40.0-44.0</th>
<th>44.0-48.0</th>
<th>48.0-52.0</th>
<th>52.0-56.0</th>
<th>56.0-60.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>12.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>16.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>20.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>24.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>28.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>32.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>36.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>40.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
AXLE LOAD VERSUS AXLE PLACEMENT

TANDEM SPACING IS 40 INCHES OR LESS
(LAASHC CATEGORIES)

1, 2 AND 3 INDICATE SINGLE, BITANDEM AND TRITANDEM AXLES

<table>
<thead>
<tr>
<th>AXLE LOAD</th>
<th>UNDER 12 TONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNDER 1</td>
<td>39</td>
</tr>
<tr>
<td>1 - 3</td>
<td>361</td>
</tr>
<tr>
<td>3 - 5</td>
<td>0</td>
</tr>
<tr>
<td>5 - 7</td>
<td>0</td>
</tr>
<tr>
<td>7 - 9</td>
<td>0</td>
</tr>
<tr>
<td>9 - 11</td>
<td>0</td>
</tr>
<tr>
<td>11 - 13</td>
<td>0</td>
</tr>
<tr>
<td>13 - 15</td>
<td>0</td>
</tr>
<tr>
<td>15 - 17</td>
<td>0</td>
</tr>
<tr>
<td>17 - 19</td>
<td>0</td>
</tr>
<tr>
<td>19 - 21</td>
<td>0</td>
</tr>
<tr>
<td>21 - 23</td>
<td>0</td>
</tr>
<tr>
<td>23 - 25</td>
<td>0</td>
</tr>
<tr>
<td>25 - 27</td>
<td>0</td>
</tr>
<tr>
<td>27 - 29</td>
<td>0</td>
</tr>
<tr>
<td>29 - 31</td>
<td>0</td>
</tr>
<tr>
<td>31 - 33</td>
<td>0</td>
</tr>
<tr>
<td>33 - 35</td>
<td>0</td>
</tr>
<tr>
<td>OVER 35</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL AXLES</td>
<td>420</td>
</tr>
<tr>
<td>TOTAL VEHICLES</td>
<td>210</td>
</tr>
<tr>
<td>MEAN AXLE WEIGHT</td>
<td>1.5</td>
</tr>
<tr>
<td>STANDARD DEVIATION</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Fig. 12. (Con't) Typical output data from intermediate tapes.
### GROSS OPERATING WEIGHT VERSUS OPERATING SPEED

<table>
<thead>
<tr>
<th>GROSS OPERATING WEIGHT (KIPS)</th>
<th>UNDER 20</th>
<th>20-40</th>
<th>40-50</th>
<th>50-55</th>
<th>55-60</th>
<th>60-65</th>
<th>65-70</th>
<th>70-80</th>
<th>80-90</th>
<th>90+</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNDER 4</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>29</td>
<td>13</td>
<td>29</td>
<td>11</td>
<td>212</td>
</tr>
<tr>
<td>4 - 10</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>16</td>
<td>31</td>
<td>10</td>
<td>0</td>
<td>75</td>
</tr>
<tr>
<td>10 - 15</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>16</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>34</td>
</tr>
<tr>
<td>15 - 20</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>20 - 22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>22 - 24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>24 - 26</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>26 - 28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>28 - 30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>30 - 32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>32 - 34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>34 - 36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>36 - 38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>38 - 40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>40 - 45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>45 - 50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>50 - 55</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>55 - 60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>60 - 65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>65 - 70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>70 - 75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>75 - 80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>80 - 85</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>85 - 90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>90 - 95</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OVER 95</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL VEHICLES</td>
<td>2</td>
<td>9</td>
<td>20</td>
<td>19</td>
<td>16</td>
<td>10</td>
<td>62</td>
<td>198</td>
<td>49</td>
<td>15</td>
<td>400</td>
</tr>
</tbody>
</table>

**Mean Gross Weight:**

| Weight | 2.7   | 11.5  | 16.8  | 14.4  | 11.9  | 11.6  | 9.0   | 8.7   | 8.5   | 11.1 | ----  |

**Standard Deviation:**

| Deviation | 1.6   | 18.4  | 19.7  | 13.6  | 13.5  | 17.1  | 10.7  | 10.2  | 10.9  | 16.3 | ----  |

Fig. 12. (Con't) Typical output data from intermediate tapes.
**OPERATING SPEED VERSUS AXLE PLACEMENT**

**TANDEM SPACING IS 40 INCHES OR LESS (AASHTO CATEGORIES)**

1, 2 AND 3 INDICATE SINGLE, BITANDEM AND TRITANDEM AXLES.

### AXLE PLACEMENT

<table>
<thead>
<tr>
<th>Operating Speed (mph)</th>
<th>Under 1 Tons 110</th>
<th>110</th>
<th>111</th>
<th>120</th>
<th>111</th>
<th>1210</th>
<th>1210</th>
<th>13200</th>
<th>13111</th>
<th>13210</th>
<th>131120</th>
<th>131300</th>
<th>13300</th>
<th>14200</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNDER 20</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20 - 40</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40 - 50</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50 - 55</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>55 - 60</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>60 - 65</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>65 - 70</td>
<td>29</td>
<td>26</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>70 - 80</td>
<td>112</td>
<td>62</td>
<td>11</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>80 - 90</td>
<td>29</td>
<td>16</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OVER 90</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL VEHICLES</td>
<td>210</td>
<td>125</td>
<td>26</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MEAN SPEED</td>
<td>72.8</td>
<td>72.1</td>
<td>65.6</td>
<td>0.0</td>
<td>65.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>65.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>STANDARD DEVIATION</td>
<td>14.3</td>
<td>10.8</td>
<td>15.3</td>
<td>0.0</td>
<td>12.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>16.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### AXLE PLACEMENT

<table>
<thead>
<tr>
<th>Operating Speed (mph)</th>
<th>111111</th>
<th>121100</th>
<th>112110</th>
<th>111120</th>
<th>111120</th>
<th>112120</th>
<th>112200</th>
<th>121200</th>
<th>123200</th>
<th>132300</th>
<th>131300</th>
<th>131310</th>
<th>131120</th>
<th>OVER 6 VEHICLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNDER 20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>20 - 40</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>40 - 50</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>50 - 55</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>55 - 60</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>60 - 65</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>65 - 70</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>70 - 80</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>80 - 90</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>OVER 90</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>TOTAL VEHICLES</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>400</td>
</tr>
<tr>
<td>MEAN SPEED</td>
<td>59.8</td>
<td>45.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>400</td>
</tr>
<tr>
<td>STANDARD DEVIATION</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>400</td>
</tr>
</tbody>
</table>

Fig. 12. (Con't) Typical output data from intermediate tapes.
<table>
<thead>
<tr>
<th>Code</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td><img src="image" alt="Vehicle" /></td>
</tr>
<tr>
<td>111</td>
<td><img src="image" alt="Vehicle" /></td>
</tr>
<tr>
<td>112</td>
<td><img src="image" alt="Vehicle" /></td>
</tr>
<tr>
<td>121</td>
<td><img src="image" alt="Vehicle" /></td>
</tr>
<tr>
<td>122</td>
<td><img src="image" alt="Vehicle" /></td>
</tr>
<tr>
<td>123</td>
<td><img src="image" alt="Vehicle" /></td>
</tr>
</tbody>
</table>

Fig. 13. Computer program codes for vehicle types.
Field Tape Obtained from RODAS
P/I Program for Analysis of

APPE N D I X A
VEHICLE: PROCEDURE OPTIONS (MAIN);

1. VEHICLE: PROCEDURE OPTIONS (MAIN);
   OPEN FILE(TAPE) INPUT, FILE(DISK) OUTPUT, FILE(DISK1) OUTPUT;
   OPEN FILE(TAPE) OUTPUT;

2. DECLARE
   STORE(IO,9) BINARY(31) INITIAL(90); FIXED STATIC;

3. DECLARE
   (RECORD(21), CHECK_CHAR) INITIAL(21), LINEPRINT_CHAR(120);

4. DECLARE
   ESPEED, TOSPEED, AXLE_SPACE_TIME, NUMBER_AXLE,
   TOTAL_WEIGHT, TOTAL_WEIGHTS, WT_COUNT,
   DIFF, INVALID_VEH, TEST_VEHICLE;

5. DECLARE
   (AXLE_SPACE, INITIAL(11)), VALID_ID,
   VALID_TIME, AXLE_COUNT_GT_9; INITIAL(11));

6. DECLARE
   (WEIGHT(2), SPEED_TIME(2), HOUR(2), ID(2));

7. DECLARE
   AXLE_SPACING; FIXED STATIC;

8. DECLARE
   SPEED, HOUR, WEIGHT_MAX, AXLE_MAX;

9. DECLARE
   SPEED_MIN, INITIAL(100), AXLE_MIN; INITIAL(100); FIXED STATIC;

10. DECLARE
    SPEED_MIN, INITIAL(100), AXLE_MIN; INITIAL(100); FIXED STATIC;

11. DECLARE
    PLATFORM BINARY INITIAL(4.3), AXLE_DOUBLE;

12. DECLARE
    FSPEEDAVG, FSPEED_MAX, AVERAGE_SPACING;

13. DECLARE
    WT_MIN, AXLE_MAX, AVERAGE_SPACING;

14. DECLARE
    AXLE_SP, SPEED_MAX; FLOAT STATIC;

15. FORMAT 1: FORMAT (COLUMN(14), A, X(4), 4, X(10), A, X(9), A, X(7), A);

16. FORMAT 2: FORMAT (COLUMN(41), F(4,0), X(4), F(4,1), A, X(6), F(2,0));

17. ON ENDFILE(TAPE) BEGIN;

18. GO TO 100; END;

19. BEGIN: GET FILE(TAPE) EDIT (CHECK) (A11); IF CHECK=0/0 THEN GO TO

20. START; ELSE GO TO BEGIN;

21. START: GET FILE(TAPE) EDIT (RECORD(I) DO I=1 TO 21) (A11);

22. IF RECORD(21) = / THEN DO: VALIDITY=1; PUT FILE(SYSPRINT) SKIP(4)

23.
VEHICLE PROCEDURE OPTIONS (MAIN):

EDIT (*+++++++++++++++++++ NON-UNIFORM RECORD LENGTH
ENCOUNTERED ++++++++++++++++++) (COLUMN(27),A,A):

FILE(SYSPRINT) SKIP(4); GO TO BEGIN; END;

INVALID_CHAR_CHECK: DO K = 1 TO 20;
IF RECORD(K) = '1' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '2' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '3' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '4' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '5' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '6' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '7' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '8' THEN GO TO END_OF_CHECK;
IF RECORD(K) = '9' THEN GO TO END_OF_CHECK;
INVALID_CHAR_FOUND: GO TO STMT(K); STMT(1);STMT(2);STMT(3);
RECORD(1) = '0'; RECORD(2) = '0'; RECORD(3) = '0'; GO TO END_OF_CHECK;
STMT(4);STMT(5);STMT(6);STMT(7);STMT(8);STMT(9);STMT(10);STMT(11);STMT(12);
STMT(13);STMT(14);STMT(15);STMT(16);STMT(17);STMT(18);DO I = 11 TO 19; RECORD(I) = '0';END;
GO TO END_OF_CHECK;
RECORD(20) = '0'; VALID_ID = 1;
END_OF_CHECK:END INVALID_CHAR_CHECK;

OBTAIN VALUES:

WEIGHT(1A) = RECORD(11) || RECORD(2) || RECORD(3);
SPEED_TIME(1A) = RECORD(4) || RECORD(5) || RECORD(6) || RECORD(7);
DAY(1A) = RECORD(8) || RECORD(9) || RECORD(10);
HOUR(1A) = RECORD(11) || RECORD(12);
MINUTE(1A) = RECORD(13) || RECORD(14);
SECMIL(1A) = RECORD(15) || RECORD(16) || RECORD(17) || RECORD(18) || RECORD(19);
ID(1A) = RECORD(20);
IF ID(1A) = 0 THEN GO TO C2;
IF SPEED_TIME(1A) = 0 THEN DO; IF DAY(1A) = 0 & VALID_TIME = 0 THEN DO;
VALIDITY = 0; GO TO START; END;
SPEED = 0; VALIDITY = 1; GO TO S4; END;
SPEED = PLATFORM * 1.647 * SPEED_TIME(1A));
IF VALID_TIME = 1 THEN GO TO S3;
IF RECORD_COUNT = 1 THEN DO; RECORD_COUNT = 0; FSPEED = SPEED;
... GO TO S4; END;
GO TO S5;
S2: VALIDITY = 1;
S3: VALID_TIME = 0; VALID_TIME2 = 1;
S4: AXLE_SPACING = 0;
GO TO S6;
S5: IF VALID_TIME2 = 1 THEN DO; VALID_TIME2 = 0; VALIDITY = 1; GO TO S6; END;
IF JA = 1 THEN DO; IF HOUR(1) = HOUR(2) THEN GO TO A1;
ELSE DO; TIME_PRINT = 1; HOURPR1 = HOUR(2); HOURPR2 = HOURPR1 + 1; END;
A1: AXLE_SPACE_TIME = SECMIL(1) - SECMIL(2); GO TO A2; END;
ELSE DO; IF HOUR(1) = HOUR(2) THEN GO TO A3; ELSE DO; TIME_PRINT = 1;
HOURPR1 = HOUR(1); HOURPR2 = HOURPR1 + 1; END;
A3: AXLE_SPACE_TIME = SECMIL(2) - SECMIL(1); END;
A2: IF AXLE_SPACE_TIME <= 0 THEN AXLE_SPACE_TIME = AXLE_SPACE_TIME +
VEHICLE: PROCEDURE OPTIONS (MAIN);

141 IF FSPEED=0 THEN GO TO S6;
143 AXLE_SPACING = 1.467 / (E+3 * AXLE_SPACE_TIME * FSPEED);
144 S6: NUMBER_AXLE = NUMBER_AXLE + 1;
145 IF AXLE_COUNT_GT_9 = 1 THEN GO TO C15;
147 IF NUMBER_AXLE > 10 THEN GO TO C14;
149 IF AXLE_SPACING > 350 THEN GO TO S7; ELSE GO TO C16;
152 C50: AXLE_SPACING = 0; VALIDITY = 0; NUMBER_AXLE = 1; FSPEED = SPEED;
156 C16: IF WEIGHT(JA) > 200 THEN WEIGHT(JA) = 200;
158 IF WEIGHT(JA) > WEIGHTMAX THEN WEIGHTMAX = WEIGHT(JA);
160 IF WEIGHT(JA) < WEIGHTMIN THEN WEIGHTMIN = WEIGHT(JA);
162 IF TOTAL_WEIGHT = TOTAL_WEIGHT + WEIGHT(JA);
163 WCOUNT = WCOUNT + 1;
164 IF SPEED > SPEEDMAX THEN SPEEDMAX = SPEED;
166 IF SPEED < SPEEDMIN THEN SPEEDMIN = SPEED;
168 NUMBER_AXLE = NUMBER_AXLE + 1;
169 IF NUMBER_AXLE > 10 THEN GO TO C14;
171 IF NUMBER_AXLE = 10 THEN GO TO S11;
173 IF NUMBER_AXLE = 1 THEN GO TO S11;
175 IF NUMBER_AXLE > 1 THEN GO TO C16;
178 IF NUMBER_AXLE < NUMBER_AXLE - AXLE_SPACING;
180 TOTAL_AXLES = TOTAL_AXLES + AXLE_SPACING;
181 S11: IF AXLE_COUNT_GT_9 = 1 THEN GO TO C2;
183 STORE(Number_AXLE,1) = WEIGHT(JA);
185 STORE(Number_AXLE,2) = FSPEED;
187 STORE(Number_AXLE,3) = AXLE_SPACING;
189 STORE(Number_AXLE,4) = MINUTE(JA);
191 STORE(Number_AXLE,5) = MINUTE(JA);
193 STORE(Number_AXLE,6) = MINUTE(JA);
195 STORE(Number_AXLE,7) = MINUTE(JA);
197 STORE(Number_AXLE,8) = MINUTE(JA);
199 STORE(Number_AXLE,9) = IC(JA);
201 C2: IF JA = 1 THEN JA = 2; ELSE JA = 1; GO TO START;
203 STORE(Number_AXLE,9) = 9 THEN DO; CALL ONEAXLE; GO TO C50;
205 IF TIME_PRINT = 1 THEN CALL EOHP; GO TO C50;
207 VALID_VEHICLE = VALID_VEHICLE + 1;
209 DO I = 1 TO NUMBER_AXLE;
211 IF NUMBER_AXLE = 1 THEN GO TO C25;
213 IF STORE(I,9) = 9 THEN GO TO C30; IDPRINT = STORE(I,9);
215 IF NUMBER_AXLE = 1 THEN DO; CALL ONEAXLE; GO TO C50; END;
217 VALID_VEHICLE = VALID_VEHICLE + 1;
219 CALL PRVALID;
221 IF NUMBER_AXLE = 1 THEN CALL ONEAXLE; GO TO C50;
223 IF TIME_PRINT = 1 THEN CALL EOHP; GO TO C50;
225 IF STORE(I,9) = 9 THEN DO; TEST_VEHICLE = TEST_VEHICLE + 1;
227 C30: TEST_VEHICLE = TEST_VEHICLE + 1;
229 CALL PRVALID;
231 IF TIME_PRINT = 1 THEN CALL EOHP; GO TO C50;
233 IF STORE(I,9) = 9 THEN DO; TEST_VEHICLE = TEST_VEHICLE + 1;
235 IF TIME_PRINT = 1 THEN CALL EOHP; GO TO C50;
237 IF TIME_PRINT = 1 THEN CALL EOHP; GO TO C50;
VEHICLE :  PROCEDURE OPTIONS (MAIN);

  * (INVALID) (COLUMN(52),A,F(2,0),A);

225 CALL PRINTAX; IF TIME_PRINT = 1 THEN CALL EOHP; GO TO 250; END;
230 ELSE DO: CALL ONEAXLE; GO TO 250; END;
234 ONEAXLE :  PROCEDURE;
235 INVALIDVEH=INVALIDVEH+1; PUT FILE(DISK) SKIP(5) EDIT
236 ('INVALID VEHICLE NO. ',INVALIDVEH) (COLUMN(55),A,F(4,0));
237 CALL PRINTAX; IF TIME_PRINT = 1 THEN CALL EOHP;
240 END ONEAXLE:

/****** TAKING CARE OF MORE THAN 9 AXLES ON A VEHICLE *************/

241 C14:AXLE_COUNT_GT_9 = 1;
242 PUT FILE(DISK) SKIP(4) EDIT ('INVALID BECAUSE OF INABILITY TO ',
243 'DISTINGUISH BETWEEN VEHICLES') (COLUMN(36),A,A);
244 PUT FILE(DISK) SKIP(2) EDIT ('AXLE: NUMBER', 'WEIGHT', 'SPEED',
245 'SPACING', 'I.D') (FORM(RMAT)); PUT FILE(DISK) SKIP(2);
246 DO I=1 TO 10: WPRINT = STORE(I,1)*1.E-1; SPEEDPR = STORE(I,1)*1.E-1;
247 AXLESPPR = STORE(I,3)*1.E-1;
248 PUT FILE(DISK) SKIP EDIT (NUMBER_AXLE,WPRINT, ' KIPS',SPEEDPR,' MPH',
249 AXLESPPR,' FEET',STORE(I,9)) (FORM(RMAT)); END;

250 C15: IF AXLE_SPACING > 350 THEN DO: AXLE_COUNT_GT_9 = 0; GO TO 250; END;

251 WPRINT = WEIGHT(JA) * 1; SPEEDPR = SPEED * 1; AXLESPPR = AXLE_SPACING * 1;

253 PUT FILE(DISK) SKIP EDIT (NUMBER_AXLE,WPRINT, ' KIPS',SPEEDPR, ' MPH',
255 AXLESPPR,' FEET',I.D(JA)) (FORM(RMAT));

260 IF TIME_PRINT = 1 THEN CALL EOHP; GO TO C16;

/****** LAST STEP OF THE OPERATION ***********/

262 C100: HOURPR1 = STORE(I,5); IF HOURPR1 = 24 THEN HOURPR1 = 0;
264 HOURPR2 = HOURPR1 + 1;
267 PUT FILE(SYSPRINT) PAGE EDIT (************* BEGIN VEHICLE ',
268 'DATA FOR THE LAST (PARTIAL) HOUR FROM ', HOURPR1, ' TO ',
269 HOURPR2, ' *******************') (COLUMN(47),A,A);

270 PUT FILE(SYSPRINT) SKIP(2) EDIT ('AXLE: NUMBER', 'WEIGHT', 'SPEED',
271 'SPACING', 'I.D') (COLUMN(35),A,X(9),A,X(10),A,X(9),A,X(7),A);

272 PUT FILE(SYSPRINT) SKIP; DO I=1 TO NUMBER_AXLE;
274 WPRINT = STORE(I,1)*1.E-1; SPEEDPR = STORE(I,1)*1.E-1;
276 AXLESPPR = STORE(I,3)*1.E-1;

277 PUT FILE(SYSPRINT) SKIP EDIT (I,WPRINT, ' KIPS',SPEEDPR,' MPH',
278 AXLESPPR,' FEET', STORE(I,9)) (COLUMN(41),F(4,0),X(4),
279 F(4,1),A,X(6),F(5,1),A,X(7),F(4,1),A,X(6),F(4,0)); END;

280 PUT FILE(SYSPRINT) SKIP(4) EDIT (*********** END OF LAST HOUR ',HOURPR1, ' TO ',
281 '****************************') (COLUMN(21),A,A,F(2,0),
282 A,F(2,0),A);

/****** PRINT AXLE PROCEDURE ***********/

280 PRINTAX :  PROCEDURE;
281 PUT FILE(DISK) SKIP(2) EDIT ('AXLE: NUMBER', 'WEIGHT', 'SPEED',
282 'SPACING') (COLUMN(40),A,X(5),A,X(9),A,X(7),A);
284 PUT FILE(DISK) SKIP;
285 DO I=1 TO NUMBER_AXLE;
286 WPRINT = STORE(I,1)*1.E-1;
288 SPEEDPR = STORE(I,2)*1.E-1; AXLESPPR = STORE(I,3)*1.E-1;
290 PUT FILE(DISK) SKIP EDIT (I,WPRINT, ' KIPS',SPEEDPR,' MPH',
291 AXLESPPR,' FEET') (COLUMN(46),F(4,0),X(5),F(4,1),A,X(5),
292 F(4,1),A,X(5),F(4,1),A); END;

299 END PRINTAX.
VFHLE : PROCEDURE OPTIONS (MAIN);

/****** END OF PRINT AXLE PROCEDURE******/
/****** "END OF HOUR" PRINT OUT PROCEDURE******/

EOHP : PROCEDURE; TIME_PRINT = 0;

PUT FILE(SYSPRINT) PAGE EDIT (****************** BEGIN,
  'VEHICLE DATA FOR THE HOUR ',HOURPR1, ' TO ',HOURPR2,
  '******************') (COLUMN(21),A,A,F(2,0),A,F(2,0),A);

CALL AVFLLT;

PUT FILE(SYSPRINT) SKIP(5) EDIT (****************** BEGIN,
  '**** END OF HOUR ',HOURPR1, ' TO ',HOURPR2,
  '******************') (COLUMN(21),A,A,F(2,0),A,F(2,0),A);

END EOHP;

/****** END OF "END OF HOUR" PRINT OUT PROCEDURE******/

AVERAGL : PROCEDURE;

FSPEEDAVE = TOTALSPEED*1.E-1/WTCOUNT;
FSPEEDMAXPR = FSPEEDMAX*1.E-1; FSPEEDMINPR = FSPEEDMIN*1.E-1;
AVERAGE = TOTAL_WEIGHT*1.E-1/WTCOUNT;
AVERAGESPEED = TOTALSPEED*1.E-1/WTCOUNT;
DIFF = WTCOUNT - TESTVEHICLE - INVALIDVEH;
IF DIFF < 0 THEN AVERAGESPACING = TOTALAXLESP * 1.E-1/DIFF;
IF DIFF > 0 THEN AVERAGESPACING = TOTALAXLESP * 1.E-1/DIFF;

PUT_FILE(SYSPRINT) SKIP EDIT ('NUMBER OF VALID VEHICLES FOR THIS HOUR=',VALIDVEHICLE,('STATION IDEN=',STATION IDEN,
  'IDENTIFICATION=',IDPRINT) (COLUMN(48),A,F(3,0),A,A,F(1,0)));

PUT_FILE(SYSPRINT) SKIP EDIT ('INVALID VEHICLES=',INVALIDVEH,
  'TEST VEHICLES=',TESTVEHICLE) (COLUMN(43),2(A,F(4,0)));

PUT_FILE(SYSPRINT) SKIP EDIT (NUMBER OF AXLES FOUND THIS HOUR = ',
  'WTCOUNT) (COLUMN(45),A,F(5,0)));

PUT_FILE(SYSPRINT) SKIP (2) EDIT ('MAXIMUM', 'MINIMUM', 'AVERAGE')
  (COLUMN(52),3(A,X(8))); PUT_FILE(SYSPRINT) SKIP (2) EDIT
  ('ALL AXLE SPEED',SPEEDMAXPR, 'MPH',
  SPEEDMINPR, 'MPH', AVERAGE_SPEED, 'MPH') (COLUMN(31),A,
  3(X(5),F(5,1),A));

PUT_FILE(SYSPRINT) SKIP (2) EDIT ('FIRST AXLE SPEED',FSPEEDMAXPR,
  'MPH', 'FSPEEDMINPR', 'MPH', 'FSPEEDAVE', 'MPH')
  (COLUMN(29),A,3(X(5),F(5,1),A));

PUT_FILE(SYSPRINT) SKIP (2) EDIT ('WEIGHT',WTMAXPR, 'KIPS',
  WTMINPR, 'KIPS', AVERAGEWT, 'KIPS') (COLUMN(39),A,3(X(5),F(4,1),A));

PUT_FILE(SYSPRINT) SKIP (2) EDIT ('AXLE SPACING',AXLESPMAXPR, 'FEET',
  AXLESPMINPR, 'FEET', AVERAGE_SPACING, 'FEET') (COLUMN(33),A,
  3(X(4),F(4,1),A));

PUT_FILE(SYSPRINT) SKIP (2);

CLOSE_FILE(DISK); OPEN_FILE(DISK) INPUT;
ON END_FILE riv DISK) GO TO FINISHED;

READ_DISK : SET_FILE(DISK) EDIT (LINEPRINT) (A(120)) COPY;
GO TO READ_DISK;

REVERT_END_FILE(DISK);

FINISHED : CLOSE_FILE(DISK); OPEN_FILE(DISK) OUTPUT;
CLOSE_FILE(DISK2); OPEN_FILE(DISK2) OUTPUT;
ON_END File(DISK2) GO TO DONE;

27
VEHICLE: PROCEDURE OPTIONS (MAIN):

335 PUT FILE(SYSPRINT) SKIP(4) EDIT ('GROSS','AXLE SPACING (FEET)',
  'VEHICLE NO.', 'SPEED', 'WEIGHT', 1-2 2-3 3-4 4-5',
  5-6 6-7 7-8 8-9) (COLUMN(152),A,X(18),A,COLUMN(26),
  A,A);
336 PUT FILE(SYSPRINT) SKIP;
337 ON ENDPAGE(SYSPRINT) BEGIN:
339 PUT FILE(SYSPRINT) PAGE EDIT ('GROSS','AXLE SPACING (FEET)',
  'VEHICLE NO.', 'SPEED', 'WEIGHT', 1-2 2-3 3-4 4-5',
  5-6 6-7 7-8 8-9) (COLUMN(152),A,X(18),A,COLUMN(26),
  A,A);
340 PUT FILE(SYSPRINT) SKIP(2);
341 END;
342 RD: GET FILE(DISK2) EDIT (LINEPRINT) (A(120)) COPY; GO TO RD;
344 DONE; CLOSE FILE(DISK2); OPEN FILE(DISK2) OUTPUT;
346 FSPEEDMAX=0; FSPEEDMIN=2000; FTOTALSPEED=0;
349 WTCOUNT=0; WEIGHTMAX=0; WEI GTMIN=900; TOTAL_WEIGHT=0; SPEEDMAX=0;
354 SPEEDMIN=1300; TOTALSPEED=0; AXLESPMAX=0; AXLESPMIN=400;
358 TOTALAXLESP=0; TESTVEHICLE=0; VALIDVEHICLE=0; INVALIDVEH=0;
362 END AVEFLE;
363 PKVALID: PROCEDURE;
364 WPRINT = 0.0; DO I=1 TO NUMBER_AXLE;
366 WPRINT = WPRINT + STORE(I,11) * 1.E-1; END;
368 PUT FILE(DISK2) SKIP EDIT (VALIDVEHICLE,STORE(I,2) * 1.E-1, 'MPH',
  WPRINT, ' KIPS', (STORE(I,3) * 1.E-1, 00 1=2 TO NUMBER_AXLE)
  (COLUMN(25),F(I7,0),X(5),F(5,1),A,X(3),F(4,1),A,9(X(2),F(4,1)
  1)));
369 END PKVALID;
370 FND VEHICLE;
VEHDATA : PROCEDURE OPTIONS (MAIN);

OPEN FILE(SYSPRINT) OUTPUT LINESIZE(132);
DECLARE SLASH CHAR(1);
DECLARE (GOWVSAX(30,11) INIT((33010)),
GOWVSAX(27) INIT((81010)),
GOWDSAX(11,27) INIT((129710)),
AXLDSAX(30,21) INIT((63010)),
CA INIT(0),
DAY INIT(0),
NUMOFAXLES INIT(0),
AXLENUM INITIAL(0),
DAYHOLD INIT(0),
NUMBER INIT(0),
SPO INIT(0),
IOVSAX(30,130),
GOWVSAX(1) INIT((B10101),
AXLDVSAX(30,211),
CA INIT(0),
DAY INIT(0),
NUMOFAXLES INIT(0),
AXLENUM INITIAL(0),
DAYHOLD INIT(0),
NUMBER INIT(0),
SPO INIT(0),
IOVSAX(30,130),
GOWVSAX(1) INIT((B10101),
NUMVEH INIT(0),
SPEED INIT(0),
TA INIT(0),
SPACING INIT(0),
MINLDAASH01291 INIT(29199999),
MINLOADKY129 INIT(29199999));
DECLARE MAXMINA ENTRY (FIXED BINARY (31)),
MAXMINK ENTRY (FIXED BINARY (31)),
GOWCHK ENTRY (FIXED BINARY (31)),
SPDCHK ENTRY (FIXED BINARY (31)),
AXLOAD ENTRY (FLOAT BINARY(16), FIXED BINARY (31)),
ADDAXL ENTRY (FIXED BIN(31), FIXED BIN(31), FLOAT BIN(16)),
ADAXLAA ENTRY (FIXED BIN(31), FIXED BIN(31), FLOAT BIN(16)),
ADDVHAA ENTRY (FIXED BIN(31), FIXED BIN(31), FIXED BIN(31)),
TANDAA ENTRY (FLOAT BIN(16)),
ADDVHKY ENTRY (FIXED BINARY (31)),
PRGWOS ENTRY (FIXED BIN(31)),
PRGWAX1 ENTRY (FIXED BIN (31)),
PRGWAX2 ENTRY (FIXED BIN (31)),
PRLDAX1 ENTRY (FIXED BIN (31)),
PRLDAX2 ENTRY (FIXED BIN (31)),
PRLDAX3 ENTRY (FIXED BIN (31)),
PRLDAX4 ENTRY (FIXED BIN (31)),
PROSAX1 ENTRY (FIXED BIN (31)),
PROSAX2 ENTRY (FIXED BIN (31)),
ADDVHAA ENTRY (FIXED BINARY (31));
IF 10=0 THEN GO TO BEGIN;
IF DAY=0 THEN GO TO BEGIN;
IF SLASH ^= '/' THEN DO; A1:PUT FILE(SYSPRINT) SKIP(5) EDIT
VEHDATA : PROEDURE OPTIONS (MAIN):

('+++++++++++++++++++++++++++++ 25 NUMBERS WERE NOT FOUND BETWEEN SLASHES +++++++++++++++++++')(:A2:GET FILE (TAPE) EDIT (SLASH)(A11): IF SLASH = ' ' THEN GO TO A2; A3: GET FILE(TAPE) EDIT (WEIGHT,SPEED,SPACING,DAY,AXLENUM,ID,SLASH)(F(3,1),F(4,1),F(5,1),F(3,0),X(9),2 F(1,0),A(11)): IF SLASH = ' ' THEN GJ TO A1; IF AXLENUM = 1 THEN GO TO A3;

GO TO 62; END;

IF RECORD = 0 THEN DO; RECORD=1; DAYHOLD=DAY; IDHOLD=ID; GO TO 63; END;

IF ID = IDHOLD THEN DO; CALL SUMMARY; IDHOLD = ID; DAYHOLD=DAY; GO TO 63; END;

IF DAY = DAYHOLD THEN DO; CALL SUMMARY; DAYHOLD=DAY; GO TO 63; END;

G1: IF AXLENUM = 1 THEN
G2: DO; IF NUMBER=1 THEN DO; STORE(1,1)=WEIGHT; STORE(1,2)=SPEED; STORE(1,3)=SPACING; VEHWT = WEIGHT;

G3: NUMBER=AXLENUM; STORE(NUMBER,1) = WEIGHT; STORE(NUMBER,2)=SPEED; STORE(NUMBER,3)=SPACING; VEHWT = VEHWT + WEIGHT; NUMOFAXLES = NUMOFAXLES + 1; GO TO BEGIN;

G4: VEHWT=O.0; GO TO 63;

SUMMARY : PROCEDURE;

PUT FILE(SYSPRINT) PAGE LINE(2) EDIT ('** VEHICLE DATA FOR DAY **',DAYHOLD, (** VEHICLE DATA FOR DAY **',DAYHOLD)(COLUMN(15),9 A,F(3,0),X(1),8 A);

PUT FILE(SYSPRINT) SKIP(2) EDIT ('STATION IDENTIFICATION = ',IDHOLD)(COLUMN(53),A,F(1,0));

PUT FILE(SYSPRINT) SKIP(2) EDIT ('NUMBER OF AXLES FOR THIS DAY = ',NUMOFAXLES)(COLUMN(48),A,F(1,0));

PUT FILE(SYSPRINT) SKIP(2) EDIT ('NUMBER OF VEHICLES FOR THIS DAY = ',NUMVEH)(COLUMN(47),2 A,F(5,0));

PUT FILE(SYSPRINT) SKIP(4) EDIT {

SUMMARY : PROCEDURE;

*/

'}
VEHDATA : PROCEDURE OPTIONS (MAIN);

29-31 1024 58-62 9,59'

NOTE: KENTUCKY DOES NOT IDENTIFY TANDEM AXLES SEPARATELY.
FOR PURPOSES OF COMPUTATION.

THE FACTORS USED BY AASHO RELATE TO TRUCK AXLES.
ADDITION, TWO-AXLE, FOUR TIRED VEHICLES ARE ASSUMED.

TO CONTRIBUTE 0.0002 EAL TO A.PER.VEHICLE.

SINGLE AXLE, AASHO FACTORS RELATE TO FLEXIBLE PAVEMENTS HAVING A TERMINAL SERVICEABILITY INDEX OF 2.5.
AND A STRUCTURAL NUMBER OF 5.

(COLUMN(38),A,SKIP(2),8(COLUMN(36),A),SKIP(2),8(COLUMN(36),A));

PUT FILE(SYSPRINT). SKIP(4) EDIT {

1, 2 AND 3 INDICATE SINGLE, TANDEM AND TRITANDEM AXLES.

0 0 0 110 ',
0 0 0 111 ',
0 0 0 111 ',
0 0 0 112 ',
0 0 0 112 ',
0 0 0 130 ',

(COLUMN(39),A,SKIP(2),7 (COLUMN(53),A));

CALL GWVSOSC;
CALL GWAXC;
CALL OSVSAXC;
CALL LDAXCA;
CALL LDAXCK;
CALL EWLSC;
OSVSAX=0; SGDAX=0; SGDSAX=0; SGDOSAX=0; AXLDSAX=0; AXLDSAX=0;
NUMOFAXLES=0; NUMVEH=0; SGDAX=0; SDLAX=0; SODLAX=0; SODLAXA=0;
MAXLDSAX0=0; MEANLDAASH0=0; SDAASH0=0; MAXLOADKY=0;
MINLDAASH=99999; MINLOADKY=99999; MEANLOADKY=0; SDKY=0;
GOWVSAX=0; GOWVSOS=0;

END SUMMARY;

MAXIMA : PROCEDURE (CA);
DECLARE CA FIXED BINARY (31);
IF TA < MINLDAASHCA THEN MINLDAASHCA = TA;
IF TA > MAXLDAASHCA THEN MAXLDAASHCA = TA;
MEANLDAASHCA = MEANLDAASHCA + TA ** 2;
TA = 0.0;
END MAXIMA;

MAXMIN : PROCEDURE( CA);
DECLARE CA FIXED BINARY (31);
IF TK < MINLOADKICA THEN MINLOADKICA = TK;
IF TK > MAXLOADKICA THEN MAXLOADKICA = TK;
MEANLOADKICA = MEANLOADKICA + TK;
SDKYICA = SDKYICA + TK ** 2;
TK = 0.0;
END MAXMIN;

GOWCHK:PROCEDURE (W);
DECLARE W FIXED BINARY (31);
IF VEHWT < 4.0 THEN DO; W=1; GO TO FINISH; END;
IF VEHWT<=4.0 & VEHWT<=10. THEN DO; W = 2; GO TO FINISH; END;
IF VEHWT<=10. & VEHWT<=15. THEN DO; W = 3; GO TO FINISH; END;
IF VEHWT<=15. & VEHWT<20. THEN DO; W = 4; GO TO FINISH; END;
IF VEHWT<=20. & VEHWT<22. THEN DO; W = 5; GO TO FINISH; END;
IF VEHWT<=22. & VEHWT<24. THEN DO; W = 6; GO TO FINISH; END;
VEHDATA: PROCEDURE OPTIONS (MAIN):  

190 IF VEHWT>24. & VEHWT<26. THEN DO; W= 7; GO TO FINISH; END;  
195 IF VEHWT>25. & VEHWT<28. THEN DO; W= 8; GO TO FINISH; END;  
200 IF VEHWT>28. & VEHWT<30. THEN DO; W= 9; GO TO FINISH; END;  
205 IF VEHWT>30. & VEHWT<32. THEN DO; W= 10; GO TO FINISH; END;  
210 IF VEHWT>32. & VEHWT<34. THEN DO; W= 11; GO TO FINISH; END;  
215 IF VEHWT>34. & VEHWT<36. THEN DO; W= 12; GO TO FINISH; END;  
220 IF VEHWT>36. & VEHWT<38. THEN DO; W= 13; GO TO FINISH; END;  
225 IF VEHWT>38. & VEHWT<40. THEN DO; W= 14; GO TO FINISH; END;  
230 IF VEHWT>40. & VEHWT<45. THEN DO; W= 15; GO TO FINISH; END;  
235 IF VEHWT>45. & VEHWT<50. THEN DO; W= 16; GO TO FINISH; END;  
240 IF VEHWT>50. & VEHWT<55. THEN DO; W= 17; GO TO FINISH; END;  
245 IF VEHWT>55. & VEHWT<60. THEN DO; W= 18; GO TO FINISH; END;  
250 IF VEHWT>60. & VEHWT<65. THEN DO; W= 19; GO TO FINISH; END;  
255 IF VEHWT>65. & VEHWT<70. THEN DO; W= 20; GO TO FINISH; END;  
260 IF VEHWT>70. & VEHWT<75. THEN DO; W= 21; GO TO FINISH; END;  
265 IF VEHWT>75. & VEHWT<80. THEN DO; W= 22; GO TO FINISH; END;  
270 IF VEHWT>80. & VEHWT<85. THEN DO; W= 23; GO TO FINISH; END;  
275 IF VEHWT>85. & VEHWT<90. THEN DO; W= 24; GO TO FINISH; END;  
280 IF VEHWT>90. & VEHWT<95. THEN DO; W= 25; GO TO FINISH; END;  
285 W=26;  
286 FINISH: END GOWCHK;  
287 SPDCHK:PROCEDURE(SPD);  
288 DECLARE SPD FIXED BINARY (31);  
289 IF STORE(I,2)< 20.0 THEN DO; SPD=1; GO TO FINISH; END;  
294 IF STORE(I,2)>=20.& STORE(I,2)<40. THEN DO; SPD=2; GO TO FINISH; END;  
299 IF STORE(I,2)>=40.& STORE(I,2)<50. THEN DO; SPD=3; GO TO FINISH; END;  
304 IF STORE(I,2)>=50.& STORE(I,2)<55. THEN DO; SPD=4; GO TO FINISH; END;  
309 IF STORE(I,2)>=55.& STORE(I,2)<60. THEN DO; SPD=5; GO TO FINISH; END;  
314 IF STORE(I,2)>=60.& STORE(I,2)<65. THEN DO; SPD=6; GO TO FINISH; END;  
319 IF STORE(I,2)>=65.& STORE(I,2)<70. THEN DO; SPD=7; GO TO FINISH; END;  
324 IF STORE(I,2)>=70.& STORE(I,2)<80. THEN DO; SPD=8; GO TO FINISH; END;  
329 IF STORE(I,2)>=80.& STORE(I,2)<90. THEN DO; SPD=9; GO TO FINISH; END;  
334 SPD=10;  
335 FINISH: END SPDCHK;  
336 AXLOAD : PROCEDURE (LOAD, AL);  
337 DECLARE LOAD FLOAT BIN(161), AL FIXED BINARY .131;  
342 IF LOAD < 1.0 THEN DO; AL=1; LOADKY=0.; LDAASHO=0.; GO TO FINISH; END;  
344 IF LOAD>=1.0& LOAD<3.0 THEN DO; AL=2; LOADKY=0.0; LDAASHO=2.0E-4;  
350 GO TO FINISH; END;  
352 IF LOAD>=3.0& LOAD<5.0 THEN DO; AL=3; LOADKY=0.; LDAASHO=2.0E-4;  
357 GO TO FINISH; END;  
359 IF LOAD>=5.0& LOAD<7.0 THEN DO; AL=4; LOADKY=0.; LDAASHO=1.0E-2;  
364 GO TO FINISH; END;  
366 IF LOAD>=7.0& LOAD<9.0 THEN DO; AL=5; LOADKY=0.; LDAASHO=3.0E-2;  
371 GO TO FINISH; END;  
373 IF LOAD>=9.0& LOAD<11. THEN DO; AL=6; LOADKY=1.00; LDAASHO=.09;  
378 GO TO FINISH; END;  
380 IF LOAD>11.& LOAD<13. THEN DO; AL=7; LOADKY=.09; LDAASHO=1.9; GO TO FINISH; END;  
384 IF LOAD>13.& LOAD<15. THEN DO; AL=8; LOADKY=.40; LDAASHO=.36; GO TO FINISH; END;  
388 IF LOAD>15.& LOAD<17. THEN DO; AL=9; LOADKY=.80; LDAASHO=.62; GO TO FINISH; END;  
392 IF LOAD>17.& LOAD<19. THEN DO; AL=10; LOADKY=.16; LDAASHO=1.00; GO TO FINISH; END;  
396 IF LOAD>19.& LOAD<21. THEN DO; AL=11; LOADKY=32.; LDAASHO=1.51; GO TO FINISH; END;
VEHDATA : PROCEDURE OPTIONS {MAIN};

415 IF LOAD = 21. & LOAD < 23. THEN DO; AL = 12; LOAD = 64.1
419 LDA = 9.218; GO TO FINISH; END:
422 IF LOAD = 23. & LOAD < 25. THEN DO; AL = 13; LOAD = 128.1
426 LDA = 9.303; GO TO FINISH; END:
429 IF LOAD = 25. & LOAD < 27. THEN DO; AL = 14; LOAD = 256.1
433 LDA = 9.409; GO TO FINISH; END:
436 IF LOAD = 27. & LOAD < 29. THEN DO; AL = 15; LOAD = 512.1
440 LDA = 9.539; GO TO FINISH; END:
443 IF LOAD = 29. & LOAD < 31. THEN DO; AL = 16; GO TO A; END:
446 IF LOAD = 31. & LOAD < 33. THEN DO; AL = 17; GO TO A; END:
449 IF LOAD = 33. & LOAD < 35. THEN DO; AL = 18; GO TO A; END:
458 IF AL = 19.1 : LOAD = 1024.; LDA = 9.697;
461 FINISH : END AXLOAD;
462 ADDAXL : PROCEDURE (CAA, CAD, LD);
466 DECLARE (CAA, CAD) FIXED BINARY (31); LD FLOAT BINARY (16);
469 ADDAXLAA : PROCEDURE (CAA, CAD, LD);
470 DECLARE (CAA, CAD) FIXED BINARY (31); LD FLOAT BINARY (16);
472 AXDLVSAAX(AAA, CAAD) = AXDLVSAAX(AAA, CAAD) + 1;
475 AXDLVSAAX(AAA, CA, 20 ) = AXDLVSAAX(AAA, 20 ) + 1;
476 SOLDAX(1, CAA) = SDLDAX(1, CAA) + LD;
479 END ADDAXL;
496 TANDAA : PROCEDURE (LOAD);
499 DECLARE LOAD FLOAT BINARY (16);
503 IF LOAD < 10.0 THEN DO; LDA = 9.0; GO TO F; END;
506 IF LOAD >= 10.0 & LOAD < 14.0 THEN DO; LDA = 9.01; GO TO F; END;
509 IF LOAD = 14.0 & LOAD < 18.0 THEN DO; LDA = 9.015; GO TO F; END;
512 IF LOAD = 18.0 & LOAD < 22.0 THEN DO; LDA = 9.122; GO TO F; END;
515 IF LOAD >= 22.0 & LOAD < 26.0 THEN DO; LDA = 9.260; GO TO F; END;
523 IF LOAD = 26.0 & LOAD < 30.0 THEN DO; LDA = 9.501; GO TO F; END;
526 IF LOAD = 30.0 & LOAD < 34.0 THEN DO; LDA = 9.810; GO TO F; END;
533 IF LOAD = 34.0 & LOAD < 38. THEN DO; LDA = 1.381; GO TO F; END;
536 IF LOAD = 38.0 & LOAD < 42. THEN DO; LDA = 2.080; GO TO F; END;
543 IF LOAD >= 42.0 & LOAD < 46. THEN DO; LDA = 3.080; GO TO F; END;
546 IF LOAD = 46.0 & LOAD < 50. THEN DO; LDA = 4.170; GO TO F; END;
553 IF LOAD >= 50.0 & LOAD < 54. THEN DO; LDA = 5.630; GO TO F; END;
VEHDATA : PROCEDURE OPTIONS (MAIN):

558 IF LOAD >= 54.0 & LOAD < 58. THEN DO; LDAASHO = 7.41; GO TO F; END;
563 LDAASHO = 9.59;
564 F : END TANDAA;
565 ADDVHKY : PROCEDURE (CA);
566 DECLARE CA FIXED BINARY (31);
567 AXLDSAXA(CA+21) = AXLDSAXA(CA+21) + 1;
568 END ADDVHKY;
569 ADDVHAA : PROCEDURE (CA);
570 DECLARE CA FIXED BINARY (31);
571 AXLDSAXAA(CA+21) = AXLDSAXAA(CA+21) + 1;
572 END ADDVHAA;
573 AXLE2L : PROCEDURE:
574 CALL ADDVEH(1,SPD,WJ); CALL ADDVHKY(1); CALL ADDVHAA(1);
577 DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);
579 CALL ADDAXL(I,AL,STORE(I,1)); TA=TA*LDAASHO; TK=TK+LOADKY;
582 CALL ADAXLA(I,AL,STORE(I,1)); END;
584 CALL MAXMINA(I); CALL MAXMINA(I);
585 END AXLE2L;
587 AXLE2H : PROCEDURE;
578 CALL ADDVEH(2,SPD,WJ); CALL ADDVHKY(2); CALL ADDVHAA(2);
591 DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);
593 CALL ADDAXL(I,AL,STORE(I,1)); TA=TA*LDAASHO; TK=TK+LOADKY;
596 CALL ADAXLA(I,AL,STORE(I,1)); END;
598 CALL MAXMINA(2); CALL MAXMINA(2);
600 END AXLE2H;
601 AXLE3 : PROCEDURE;
602 IF STORE(3,3) <= 3.33 THEN DO; CALL ADDVHAA(4); CALL ADDVEH(4,SPD,
605 WJ; CALL AXLOAD(STORE(1,1),AL); TA=TA*LDAASHO;
608 DWT = STORE(2,1) + STORE(3,1); CALL ADAXLA4(AL,STORE(1,1));
610 CALL AXLOAD(DWT,AL); CALL ADAXLA4(AL,DWT); CALL TANDAA(DWT);
613 TA=TA*LDAASHO; CALL MAXMINA(4); RETURN; END;
617 CALL ADDVEH3,SPD,WJ; CALL ADDVHAA(3); DO I=1 TO 3; CALL AXLOAD
621 (STORE(I,1),AL); CALL ADAXLA3(AL,STORE(I,1)); TA=TA*LDAASHO;
623 END; CALL MAXMINA(3);
625 IF STORE(3,3) > 10.0 THEN DO; CALL ADDVHKY(3); DO I=1 TO 3; CALL
630 AXLOAD(STORE(I,1),AL); CALL ADAXL3(AL,STORE(I,1));
631 TK=TK+LOADKY; END; CALL MAXMINA(3); RETURN; END;
636 CALL ADDVHKY(4); CALL AXLOAD(STORE(1,1),AL); TK=TK+LOADKY;
639 CALL ADDAXL4(AL,STORE
640 (1,1)); DWT = STORE(2,1) + STORE(3,1); CALL AXLOAD(DWT,AL);
642 CALL ADDAXL4(AL,DWT); TR=TR+LOADKY; CALL MAXMINA(4);
643 END AXLE3;
646 AXLE4 : PROCEDURE;
647 IF STORE(3,3) <= 3.33 & STORE(4,3) <= 3.33 THEN DO; CALL ADDVHAA
650 (6); CALL ADDVEH6,SPD,WJ; CALL AXLOAD(STORE(1,1),AL); CALL
653 ADAXLA6(AL,STORE(1,1)); TA=TA=LDAASHO; DWT = STORE(2,1) +
655 STORE(3,1) + STORE(4,1); CALL AXLOAD(DWT,AL); CALL ADAXLA
657 (8,AL,DWT); CALL TANDAA(DWT); TA=TA=LDAASHO; CALL MAXMINA(6);
660 RETURN; END;
662 IF STORE(3,3) <= 3.33 THEN DO; CALL ADDVHAA(6); CALL ADDVEH6,SPD,
666 WJ; CALL AXLOAD(STORE(1,1),AL); TA=TA=LDAASHO; CALL
669 ADAXLA6(AL,STORE(1,1)); CALL AXLOAD(STORE(4,1),AL);
670 CALL ADAXLA6(AL,STORE(4,1)); TA=TA=LDAASHO; DWT =
673 STORE(2,1) + STORE(3,1); CALL AXLOAD(DWT,AL); CALL TANDAA
675 (DWT); CALL ADDAXL6(AL,DWT); TA=TA=LDAASHO; CALL MAXMINA(6);
678 RETURN; END;
680 IF STORE(4,3) <= 3.33 THEN DO; CALL ADDVHAA(7); CALL ADDVEH(7,SPD,
PROCEDURE OPTIONS (MAIN);

CALL AXLOAD(STORE(1,1),AL); TA=TA+LOADASHO; CALL ADDAXLAA(7,AL); CALL AXLOAD(STORE(2,1),AL); CALL AXLOAD(7,AL,STORE(1,1)); TA=TA+LOADASHO; DWT = STORE(3,1) + STORE(4,1) + STORE(5,1); CALL AXLOAD(14,AL,DWT); TA=TA+LOADASHO; CALL MAXMINA(14); RETURN;

IF STORE(3,3) <= 10.0 & STORE(4,3) <= 10.0 THEN DO; CALL ADDVHKY(14); CALL AXLOAD(15,AL,STORE(1,1)); TA=TA+LOADASHO; DWT = STORE(3,1) + STORE(4,1) + STORE(5,1); CALL AXLOAD(15,AL,DWT); TA=TA+LOADASHO; CALL MAXMINA(15); RETURN; END;

END AXLE4;

AXLE5 : PROCEDURE;

CALL ADDVHAAS(15); CALL ADDVHKY(5); DO 1=1 TO 4; CALL AXLOAD(16,AL,STORE(1,1)); TA=TA+LOADASHO; END;

CALL ADDVHAA(15); CALL ADDVHKY(6); DO 1=1 TO 4; CALL AXLOAD(16,AL,STORE(1,1)); TA=TA+LOADASHO; END;

CALL AXLOAD(17,AL,STORE(1,1)); TA=TA+LOADASHO; CALL ADDVHKY(7); CALL AXLOAD(17,AL,DWT); TA=TA+LOADASHO; CALL MAXMINA(17); RETURN;

IF STORE(3,3) <= 10.0 THEN DO; CALL ADDVHKY(6); CALL AXLOAD(16,AL,STORE(1,1)); TA=TA+LOADASHO; CALL AXLOAD(17,AL,DWT); TA=TA+LOADASHO; CALL MAXMINA(17); RETURN; END;

END AXLE5;
PROCEDURE OPTIONS (MAIN);

IF STORE(4,3) <= 3.33 THEN DO;
  CALL ADVHAA(11); CALL ADOVE(11, SPD, W); DO I=1 TO 2; CALL
  AXLOAD(STORE(1,1), AL); CALL ADDAXLAA(11, AL, STORE(1,1)); TA=
  TA + LDAASHO; END; DWT=STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT,
  AL); CALL ADDAXLAA(11, AL, DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;
  CALL AXLOAD(STORE(5,1), AL); CALL ADDAXLAA(11, AL, STORE(5,1));
  TA=TA+LDAASHO; CALL MAXIMA(11); RETURN; END;

IF STORE(3,3) <= 3.33 THEN DO;
  CALL ADVHAA(10); CALL ADOVE(10, SPD, W); CALL AXLOAD(STORE(1,1),
  AL); CALL ADDAXLAA(10, AL, STORE(1,1)); TA=TA+LDAASHO; DWT=
  STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT, AL); CALL ADDAXLAA(10,
  AL, DWT); CALL TANDAA(DWT); TA=TA+LDAASHO; DO I=4 TO 5; CALL
  AXLOAD(STORE(1,1), AL); CALL ADDAXLAA(10, AL, STORE(1,1));
  TA=TA+LDAASHO; END; CALL MAXIMA(10); RETURN; END;

CALL ADDVAA(9); CALL ADOVE(9, SPD, W); DO I=1 TO 5; CALL AXLOAD
  (STORE(1,1), AL); CALL ADDAXLAA(9, AL, STORE(1,1)); TA=TA+LDAASHO;
  END; CALL MAXIMA(9);

IF STORE(4,3) <= 10.0 & STORE(5,3) <= 10.0 THEN DO;
  CALL ADVHVKY(14); DO I=1 TO 2; CALL AXLOAD(STORE(1,1), AL);
  CALL ADDAXL(14, AL, STORE(1,1)); TK=TK+LOADKY;
  END; DWT=STORE(3,1)+STORE(4,1) + STORE(5,1); CALL AXLOAD(DWT, AL);
  CALL ADDAXLAA(14, AL, DWT); TA=TA+LDAASHO; CALL MAXIMA(14); RETURN;
  END;

IF STORE(3,3) <= 10.0 & STORE(4,3) <= 10.0 THEN DO;
  CALL ADDAXLAA(13); CALL AXLOAD(STORE(1,1), AL); CALL AXLOAD(13,
  AL, STORE(1,1)); TK=TK+LOADKY;
  DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL
  AXLOAD(DWT, AL); CALL ADDAXLAA(13, AL, DWT); TK=TK+LOADKY;
  CALL AXLOAD(STORE(5,1), AL); CALL ADDAXLAA(13, AL, STORE(5,1));
  TK=TK+LOADKY; CALL AXLOAD(STORE(5,1), AL); CALL MAXIMA(13); RETURN;
  END;

IF STORE(3,3) <= 10.0 & STORE(5,3) <= 10.0 THEN DO;
  CALL ADDAXLAA(12); CALL AXLOAD(STORE(1,1), AL); CALL AXLOAD(12,
  AL, STORE(1,1)); TK=TK+LOADKY;
  END; DWT=STORE(4,1)+STORE(5,1); CALL AXLOAD(DWT, AL); CALL ADDAXLAA(12,
  AL, DWT); TK=TK+LOADKY; CALL MAXIMA(12); RETURN; END;

IF STORE(4,3) <= 10.0 THEN DO;
  CALL ADVHVKY(11); DO I=1 TO 2; CALL AXLOAD(STORE(1,1), AL); CALL
  ADDAXL(11, AL, STORE(1,1)); TK=TK+LOADKY;
  END; DWT=STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT, AL); TK=TK+LOADKY;
  CALL ADDAXLAA(11, AL, DWT); CALL AXLOAD(STORE(5,1), AL); CALL
  ADDAXLAA(11, AL, STORE(5,1)); TK=TK+LOADKY;
  CALL MAXIMA(11); RETURN; END;

IF STORE(3,3) <= 10.0 THEN DO;
  CALL ADVHVKY(10); CALL AXLOAD(STORE(1,1), AL); CALL ADDAXL(10,
  AL, STORE(1,1)); TK=TK+LOADKY;
  END; DWT=STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT, AL); TK=TK+LOADKY;
  CALL ADDAXLAA(10, AL, DWT); CALL AXLOAD(STORE(5,1), AL); CALL
  ADDAXLAA(10, AL, STORE(5,1)); TK=TK+LOADKY; CALL MAXIMA(10); RETURN;
  END;

IF STORE(3,3) <= 10.0 THEN DO;
  CALL ADDVAA(9); CALL ADOVE(9, SPD, W); DO I=1 TO 5; CALL AXLOAD
  (STORE(1,1), AL); CALL ADDAXLAA(9, AL, STORE(1,1)); TA=TA+LDAASHO;
  END; CALL MAXIMA(9);

IF STORE(4,3) <= 10.0 THEN DO;
  CALL ADDVAA(14); DO J=1 TO 2; CALL AXLOAD(STORE(1,1), AL); CALL
  ADDAXLAA(14, AL, STORE(1,1)); TK=TK+LOADKY;
  END; DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT, AL);
  CALL ADDAXLAA(14, AL, DWT); TK=TK+LOADKY; CALL MAXIMA(14); RETURN;
  END;

IF STORE(3,3) <= 10.0 THEN DO;
  CALL ADDAXLAA(13); CALL AXLOAD(STORE(1,1), AL); CALL AXLOAD(13,
  AL, STORE(1,1)); TK=TK+LOADKY;
  DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL
  AXLOAD(DWT, AL); CALL ADDAXLAA(13, AL, DWT); TK=TK+LOADKY;
  CALL AXLOAD(STORE(5,1), AL); CALL ADDAXLAA(13, AL, STORE(5,1));
  TK=TK+LOADKY; CALL AXLOAD(STORE(5,1), AL); CALL MAXIMA(13); RETURN;
  END;
PROCEDURE OPTIONS (MAIN);

CALL ADDVHAA(11, AL); CALL ADDAXL(11, AL, STORE(1, 11)); TK=TK+LOADKY; END;

CALL MAXMINK(10); RETURN; END;

CALL ADDVHAA(9); DO I=1 TO 5; CALL AXLOAD(STORE(I, 1), AL); CALL ADDAXL(9, AL, STORE(I, 11)); TK=TK+LOADKY; END; CALL MAXMINK(9);

RETURN;

END AXLES;

AXLE6 : PROCEDURE;

IF STORE(3, 3)<3.33 & STORE(5, 3)<3.33 & STORE(6, 3)<3.33 THEN DO;

CALL ADDVHAA(25); CALL ADDVHAA(25, SPD, W); CALL AXLOAD(STORE(I, 1));

AL; CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO; DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(25); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(4, 3)<3.33 & STORE(6, 3)<3.33 THEN DO;

CALL ADDVHAA(24); CALL ADDVHAA(24, SPD, W); CALL AXLOAD(STORE(I, 1));

AL; CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO; DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(24); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(6, 3)<3.33 THEN DO;

CALL ADDVHAA(23); CALL ADDVHAA(23, SPD, W); DO I=1 TO 4 BY 3; CALL AXLOAD(STORE(I, 1), AL); CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO;

DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(23); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(5, 3)<3.33 THEN DO;

CALL ADDVHAA(22); CALL ADDVHAA(22, SPD, W); DO I=1 TO 1; CALL AXLOAD(STORE(I, 1), AL); CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO;

DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(22); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(5, 3)<3.33 THEN DO;

CALL ADDVHAA(21); CALL ADDVHAA(21, SPD, W); CALL AXLOAD(STORE(I, 1), AL); CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO;

DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(21); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(4, 3)<3.33 THEN DO;

CALL ADDVHAA(20); CALL ADDVHAA(20, SPD, W); DO I=1 TO 3; CALL AXLOAD(STORE(I, 1), AL); CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO;

DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(20); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(4, 3)<3.33 THEN DO;

CALL ADDVHAA(19); CALL ADDVHAA(19, SPD, W); DO I=1 TO 1; CALL AXLOAD(STORE(I, 1), AL); CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO;

DWT=STORE(I, 1)+STORE(I, 1); CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(19); RETURN; END;

IF STORE(3, 3)<3.33 & STORE(4, 3)<3.33 THEN DO;

CALL ADDVHAA(18); CALL ADDVHAA(18, SPD, W); DO I=1 TO 5; CALL AXLOAD(STORE(I, 1), AL); CALL AXLOAD(STORE(I, 1)); TD=TD+LOADKY;

AL; CALL AXLOAD(STORE(I, 1)); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL TANDAA(DWT); TA=TA+LOADSHO; DWT=STORE(I, 1);

CALL MAXMINK(18); RETURN; END;

37
PROCEDURE OPTIONS (MAIN):
 TK=TK+LOADKY; CALL MAXMINK(23); RETURN; END;

IF STORE(4,3)<10.0 & STORE(6,3)<10.0 THEN DO;

CALL ADDVHKY(22); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL); CALL ADDAXL(22,AL,STORE(I,1));

END; DO I=3 TO 5 BY 2; DWT=STORE(I,;

TK=TK+LOADKY; END; CALL MAXMINK(22); RETURN; END;

IF STORE(3,3)<10.0 & STORE(5,3)<10.0 THEN DO;

CALL ADDVHKY(21); CALL AXLOAD(STORE(I,1),AL); CALL ADDAXL(21,AL,STORE(I,1));

END; DO I=2 TO 4 BY 2; DWT=STORE(I,1)+STORE(I+1,1);

CALL AXLOAD(DWT,AL); TK=TK+LOADKY;

CALL ADDAXL(21,AL,DWT); END; CALL AXLOAD

(STORE(6,1),AL); CALL ADDAXL(21,AL,STORE(6,1));

TK=TK+LOADKY; CALL MAXMINK(21); RETURN; END;

IF STORE(6,3)<10.0 & STORE(5,3)<10.0 THEN DO;

CALL ADDVHKY(28); DO I=1 TO 3; CALL AXLOAD(STORE(I,1),AL);

ADDAXL(28,AL,STORE(I,1)); TK=TK+LOADKY;

END; DWT=STORE(4,1)+STORE(5,1);

STORE(6,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(28,AL,DWT);

TK=TK+LOADKY; CALL MAXMINK(28); RETURN; END;

IF STORE(5,3)<10.0 & STORE(4,3)<10.0 THEN DO;

CALL ADDVHKY(27); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);

ADDAXL(27,AL,STORE(I,1)); TK=TK+LOADKY;

END; DWT=STORE(3,1)+STORE(4,1)+STORE(5,1);

(5,1); CALL AXLOAD(DWT,AL); TK=TK+LOADKY;

CALL ADDAXL(27,AL,DWT); END;

IF STORE(6,3)<10.0 & STORE(5,3)<10.0 THEN DO;

CALL ADDVHKY(26); CALL AXLOAD(STORE(I,1),AL); CALL ADDAXL(26,AL,STORE(I,1));

DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL AXLOAD( DWT, AL); TK=TK+LOADKY;

END; DWT=STORE(4,1)+STORE(5,1);

STORE(6,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(26,AL,DWT);

TK=TK+LOADKY; CALL MAXMINK(26); RETURN; END;

IF STORE(5,3)<10.0 THEN DO;

CALL ADDVHKY(20); DO I=1 TO 4; CALL AXLOAD(STORE(I,1),AL); CALL ADDAXL(20,AL,STORE(I,1));

END; DWT=STORE(5,1)+STORE(6,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(20,AL,DWT);

TK=TK+LOADKY; CALL MAXMINK(20); RETURN; END;

IF STORE(5,3)<10.0 THEN DO;

CALL ADDVHKY(19); DO I=1 TO 3; CALL AXLOAD(STORE(I,1),AL); CALL ADDAXL(19,AL,STORE(I,1));

END; DWT=STORE(4,1)+STORE(5,1); CALL AXLOAD(DWT,AL); TK=TK+LOADKY;

END; DWT=STORE(4,1)+STORE(5,1); CALL AXLOAD(DWT,AL); TK=TK+LOADKY;

CALL ADDAXL(19,AL,DWT); CALL ADDAXL(19,AL,STORE(6,1));

TK=TK+LOADKY; CALL MAXMINK(19); RETURN; END;

IF STORE(4,3)<10.0 THEN DO;

CALL ADDVHKY(18); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL); CALL ADDAXL(18,AL,STORE(I,1));

END; DWT=STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT,AL); TK=TK+LOADKY;

CALL ADDAXL(18,AL,DWT); DO I=5 TO 6; CALL
VEHDATA : PROCEDURE OPTIONS (MAIN);

1405 AXLOAD(STORE[1,1],AL); CALL ADDAXL[10,AL,STORE[1,1]);
1406 TK=TK+LOADY; END; CALL MAXMIN(18); RETURN; END;
1411 IF STORE[3,3]<10 THEN DO;
1413 CALL ADDVHKY(17); CALL AXLOAD(STORE[1,1],AL); CALL ADDAXL117,
1416 AL,STORE[1,1]; TK=TK+LOADY;
1417 DWT=STORE[2,1]+STORE[3,1]; CALL AXLOAD(DWT,
1419 AL); TK=TK+LOADY;
1420 CALL ADDAXL117,AL,DWT; DO I=1 TO 6; CALL AXLOAD(STORE,
1423 (1,1),AL); CALL ADDAXL117,AL,STORE[1,1];
1424 TK=TK+LOADY; END; CALL MAXMIN(17); RETURN; END;
1429 CALL ADDVHKY(16); DO I=1 TO 6; CALL AXLOAD(STORE[1,1],AL); CALL
1433 ADDAXL116,AL,STORE[1,1];
1436 RETURN;
1437 END AXLE6 : PROCEDURE;
1439 CALL ADDVHAA(29); CALL ADDVHKY(29); CALL ADDVEH(29,SPD,W); DO I=1
1443 TO NUMBER; CALL AXLOAD(STORE[1,1],AL); CALL AXLOAD(29,AL,
1445 STORE[1,1]); TA=TA+LOADY; CALL ADDAXL29,AL,STORE[1,1]);
1447 TK=TK+LOADY; END; CALL MAXMIN(29); CALL MAXMIN(29);
1451 END AXLENS;
1452 GOWVSOSC : PROCEDURE;
1453 DO =1 TO 10;
1455 IF (GOWVSOS[1,27] - GOWVSOS[1,27] - 1) = 0 THEN DO; SDGOWOS(2,1)
1457 = 0; GO TO G11; END;
1459 DWT = (GOWVSOS[1,27] - SDGOWOS(2,1) - SDGOWOS(1,1) ** 2)/(1
1460 GOWVSOS[1,27] - GOWVSOS[1,27] - 1);
1462 IF DWT < 0.0 THEN DO; SDGOWOS(2,1) = -1; GO TO G11; END;
1464 SDGOWOS(2,1) = SQRT(DWT);
1466 G11: IF GOWVSOS[1,27] = 0 THEN DO; SDGOWOS[1,1] = 0; GO TO G2; END;
1468 SDGOWOS[1,1] = GOWVSOS(1,27) / GOWVSOS(1,27);
1470 GOWVSOS[1,27] = GOWVSOS(1,27) + GOWVSOS(1,27);
1472 G2: END;
1474 PUT FILE(SYSPRINT) PAGE;
1475 PUT FILE(SYSPRINT) SKIP 4 EDIT ('GROSS OPERATING WEIGHT VERSUS*,
1476 OPERATING SPEED*) (COLUMN(17),AxAL);
1477 PUT FILE(SYSPRINT) SKIP 5 EDIT ('GROSS WEIGHT- ') (Ax(31),Ax);
1478 PUT FILE(SYSPRINT) SKIP EDIT ('OVER TOTAL') (Ax(31),Ax);
1479 PUT FILE(SYSPRINT) SKIP EDIT ('KIPS') 20 20 40 40 50 50
1480 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50
1481 (Ax, Ax);
1482 PUT FILE(SYSPRINT) SKIP 0 EDIT ('*');
1483 PUT FILE(SYSPRINT) SKIP LIST ' UNDER 4 ' 13; CALL PRGWOS(11);
1484 PUT FILE(SYSPRINT) SKIP LIST 1' 4 - 10 ' 13; CALL PRGWOS(2);
1485 PUT FILE(SYSPRINT) SKIP LIST 1' 10 - 15 ' 13; CALL PRGWOS(3);
1486 PUT FILE(SYSPRINT) SKIP LIST 1' 15 - 20 ' 13; CALL PRGWOS(4);
1487 PUT FILE(SYSPRINT) SKIP LIST 1' 20 - 22 ' 13; CALL PRGWOS(5);
1488 PUT FILE(SYSPRINT) SKIP LIST 1' 22 - 24 ' 13; CALL PRGWOS(6);
1489 PUT FILE(SYSPRINT) SKIP LIST 1' 26 - 28 ' 13; CALL PRGWOS(7);
1490 PUT FILE(SYSPRINT) SKIP LIST 1' 28 - 30 ' 13; CALL PRGWOS(8);
1491 PUT FILE(SYSPRINT) SKIP LIST 1' 30 - 32 ' 13; CALL PRGWOS(9);
1492 PUT FILE(SYSPRINT) SKIP LIST 1' 30 - 32 ' 13; CALL PRGWOS(10);
VEHDATA: PROCEDURE OPTIONS (MAIN):

1501  PUT FILE(SYSPRINT) SKIP LIST ('32 - 34') CALL PRGWOS(11);
1503  PUT FILE(SYSPRINT) SKIP LIST ('34 - 36') CALL PRGWOS(12);
1505  PUT FILE(SYSPRINT) SKIP LIST ('36 - 38') CALL PRGWOS(13);
1507  PUT FILE(SYSPRINT) SKIP LIST ('38 - 40') CALL PRGWOS(14);
1509  PUT FILE(SYSPRINT) SKIP LIST ('40 - 45') CALL PRGWOS(15);
1511  PUT FILE(SYSPRINT) SKIP LIST ('45 - 50') CALL PRGWOS(16);
1513  PUT FILE(SYSPRINT) SKIP LIST ('50 - 55') CALL PRGWOS(17);
1515  PUT FILE(SYSPRINT) SKIP LIST ('55 - 60') CALL PRGWOS(18);
1517  PUT FILE(SYSPRINT) SKIP LIST ('60 - 65') CALL PRGWOS(19);
1519  PUT FILE(SYSPRINT) SKIP LIST ('65 - 70') CALL PRGWOS(20);
1521  PUT FILE(SYSPRINT) SKIP LIST ('70 - 75') CALL PRGWOS(21);
1523  PUT FILE(SYSPRINT) SKIP LIST ('75 - 80') CALL PRGWOS(22);
1525  PUT FILE(SYSPRINT) SKIP LIST ('80 - 85') CALL PRGWOS(23);
1527  PUT FILE(SYSPRINT) SKIP LIST ('85 - 90') CALL PRGWOS(24);
1529  PUT FILE(SYSPRINT) SKIP LIST ('90 - 95') CALL PRGWOS(25);
1531  PUT FILE(SYSPRINT) SKIP LIST ('OVER 95') CALL PRGWOS(26);
1533  PUT FILE(SYSPRINT) SKIP LIST ('TOTAL') CALL PRGWOS(27);
1534  DO I=1 TO 11; PUT FILE(SYSPRINT) EDIT('ORIGINAL') CALL PRGWOS(28); END;
1537  PUT FILE(SYSPRINT) SKIP LIST ('VEHICLES') CALL PRGWOS(29);
1539  PUT FILE(SYSPRINT) SKIP LIST ('MEAN GROSS') CALL PRGWOS(30);
1540  DO I=1 TO 11; PUT FILE(SYSPRINT) EDIT('') CALL PRGWOS(31); END;
1543  PUT FILE(SYSPRINT) SKIP LIST ('WEIGHT') CALL PRGWOS(32);
1544  DO I=1 TO 10; PUT FILE(SYSPRINT) EDIT (SDGOWOS(I,1), '1') (F(5,1),
1546  A); END;
1547  PUT FILE(SYSPRINT) EDIT ('----') CALL PRGWOS(33);
1548  PUT FILE(SYSPRINT) SKIP LIST ('STANDARD') CALL PRGWOS(34);
1549  DO I=1 TO 11; PUT FILE(SYSPRINT) EDIT('') CALL PRGWOS(35); END;
1552  PUT FILE(SYSPRINT) SKIP LIST ('DEVIATION') CALL PRGWOS(36);
1553  DO I=1 TO 10; PUT FILE(SYSPRINT) EDIT (SDGOWOS(I,2), '1') (F(5,1),
1555  A); END;
1556  PUT FILE(SYSPRINT) EDIT ('----') CALL PRGWOS(37);
1557  PUT FILE(SYSPRINT) SKIP(0) EDIT ('------------------------------')
1558          (A,A); END;
1559  END GWVSOS;
1560  PRGWOS : PROCEDURE(CA);
1562  DO I=1 TO 11:
1564  END PRGWOS;
1566  GDOWS : PROCEDURE;
1568  DO I=1 TO 29; N=GOWVSAX(I,27) * (GOWVSAX(I,27) - I);
1570  IF N=0 THEN GDOWSAX(2,1) = 0; GO TO G1; END;
1572  DWT = (GOWVSAX(I,27) - SODGOWAX(2,1,1,2)) / N;
1574  GDOWSAX(2,1) = SQR(DWT);
1576  G1: IF GDOWSAX(1,27)=0 THEN DO ;SODGOWAX(1,1) = 0; GO TO G2; END;
1578  GDOWSAX(1,1) = SODGOWAX(1,1) / GDOWSAX(1,27);
1580  GDOWSAX(30,27) = GDOWSAX(30,27) + GDOWSAX(1,27);
1582  G2: END;
1584  PUT FILE(SYSPRINT) PAGE;
1586  PUT FILE(SYSPRINT) SKIP(5) EDIT ('GROSS OPERATING WEIGHT VERSUS',
1588  'AXLE PLACEMENT') (COLUMN(35), A, A);
1590  PUT FILE(SYSPRINT) SKIP(0) EDIT ('(AASHO CATEGORIES)') (COLUMN(48), A);
1592  PUT FILE(SYSPRINT) SKIP(2) EDIT ('TANDEM SPACING IS 40 INCHES',
1594  'OR LESS') (COLUMN(38), A);
**VEHDATA : PROCEDURE OPTIONS (MAIN):**

1592 PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE, BITAN*,

1593 *DEM AND TRITANDE AXLES*) (COLUMN(31),A,A);

1594 PUT FILE(SYSPRINT) SKIP(5) LIST (' GROSS ',

1595 **OPERATING UNDER**, ' AXLE PLACEMENT',

1596 **T**)(A*X(35),A);

1597 PUT FILE(SYSPRINT) SKIP LIST (' WEIGHT | 2 TONS*);

1598 PUT FILE(SYSPRINT) SKIP EDIT (' KIPS ') 110 110 111 111 111 1210 1120 1300 111 1111 12110 12110 12120*,

1599 ' 13100 13100 13200 ')(A,A,A);

1600 PUT FILE(SYSPRINT) SKIP(1); DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT (****) (A); END;

1601 PUT FILE(SYSPRINT) SKIP LIST (' UNDER 4 '); CALL PRGWAX1(11);

1602 PUT FILE(SYSPRINT) SKIP LIST (' 4 - 10 '); CALL PRGWAX1(2);

1603 PUT FILE(SYSPRINT) SKIP LIST (' 10 - 15 '); CALL PRGWAX1(3);

1604 PUT FILE(SYSPRINT) SKIP LIST (' 15 - 20 '); CALL PRGWAX1(4);

1605 PUT FILE(SYSPRINT) SKIP LIST (' 20 - 22 '); CALL PRGWAX1(5);

1606 PUT FILE(SYSPRINT) SKIP LIST (' 22 - 24 '); CALL PRGWAX1(6);

1607 PUT FILE(SYSPRINT) SKIP LIST (' 24 - 26 '); CALL PRGWAX1(7);

1608 PUT FILE(SYSPRINT) SKIP LIST (' 26 - 28 '); CALL PRGWAX1(8);

1609 PUT FILE(SYSPRINT) SKIP LIST (' 28 - 30 '); CALL PRGWAX1(9);

1610 PUT FILE(SYSPRINT) SKIP LIST (' 30 - 32 '); CALL PRGWAX1(10);

1611 PUT FILE(SYSPRINT) SKIP LIST (' 32 - 34 '); CALL PRGWAX1(11);

1612 PUT FILE(SYSPRINT) SKIP LIST (' 34 - 36 '); CALL PRGWAX1(12);

1613 PUT FILE(SYSPRINT) SKIP LIST (' 36 - 38 '); CALL PRGWAX1(13);

1614 PUT FILE(SYSPRINT) SKIP LIST (' 38 - 40 '); CALL PRGWAX1(14);

1615 PUT FILE(SYSPRINT) SKIP LIST (' 40 - 45 '); CALL PRGWAX1(15);

1616 PUT FILE(SYSPRINT) SKIP LIST (' 45 - 50 '); CALL PRGWAX1(16);

1617 PUT FILE(SYSPRINT) SKIP LIST (' 50 - 55 '); CALL PRGWAX1(17);

1618 PUT FILE(SYSPRINT) SKIP LIST (' 55 - 60 '); CALL PRGWAX1(18);

1619 PUT FILE(SYSPRINT) SKIP LIST (' 60 - 65 '); CALL PRGWAX1(19);

1620 PUT FILE(SYSPRINT) SKIP LIST (' 65 - 66 '); CALL PRGWAX1(20);

1621 PUT FILE(SYSPRINT) SKIP LIST (' 70 - 75 '); CALL PRGWAX1(21);

1622 PUT FILE(SYSPRINT) SKIP LIST (' 75 - 80 '); CALL PRGWAX1(22);

1623 PUT FILE(SYSPRINT) SKIP LIST (' 80 - 85 '); CALL PRGWAX1(23);

1624 PUT FILE(SYSPRINT) SKIP LIST (' 85 - 90 '); CALL PRGWAX1(24);

1625 PUT FILE(SYSPRINT) SKIP LIST (' 90 - 95 '); CALL PRGWAX1(25);

1626 PUT FILE(SYSPRINT) SKIP LIST (' OVER 95 '); CALL PRGWAX1(26);

1627 PUT FILE(SYSPRINT) SKIP(1) LIST (' TOTAL ');

1628 DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (****) (A); END;

1629 PUT FILE(SYSPRINT) SKIP LIST (' VEHICLES '); CALL PRGWAX1(27);

1630 PUT FILE(SYSPRINT) SKIP LIST (' MEAN GROSS ');

1631 DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (****) (A); END;

1632 PUT FILE(SYSPRINT) SKIP LIST (' WEIGHT ');

1633 DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (SOGOWAX(1),I,' '(5,1),

1634 A); END;

1635 PUT FILE(SYSPRINT) SKIP LIST (' STANDARD ');

1636 DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (****) (A); END;

1637 PUT FILE(SYSPRINT) SKIP LIST (' DEVIATION ');

1638 DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (SOGOWAX(2),I,' '(5,1),

1639 A); END;

1640 PUT FILE(SYSPRINT) PAGE LINE (6) EDIT (' GROSS OPERATING WEIGHT ',

1641 'VERSUS AXLE PLACEMENT (CONTINUED FROM PRECEDING PAGE)'(COLUMN(29),A,A);

1642 PUT FILE(SYSPRINT) SKIP(2) EDIT (' TANDEM SPACING IS 40 INCHES ',

1643 'OR LESS ') (COLUMN(38),A,A);

1644 PUT FILE(SYSPRINT) SKIP EDIT ('AASHO CATEGORIES') (COLUMN(48),A);
VEHDAT : PROCEDURE OPTIONS (MAIN);

1682  PUT FILE(SYSPRINT) SKIP(2) EDIT ('1', '2' AND '3' INDICATE SINGLE, BITAN',
       'DEM AND TRITANDM AXLES' (COLUMN(13)), A, A);
1683  PUT FILE(SYSPRINT) SKIP(5) LIST (A, 'GROSS' [ ]) [ ];
1684  PUT FILE(SYSPRINT) SKIP EDIT (A, 'WEIGHT' [ ]), 'AXLE PLACEMENT',
       (A, X(35), A);
1685  PUT FILE(SYSPRINT) SKIP EDIT ('T', 'WEIGHT [1], 'TOTAL' [1]),
       (A, X(93), A);
1686  PUT FILE(SYSPRINT) SKIP EDIT ('KIPS'), 111111 121111 1121110 111111,
       1211 111120 122100 121200 121200 123000 123000 131100 131100,
       131100 OVER 6 VEHICLES) (A, A, A);
1687  PUT FILE(SYSPRINT) SKIP(PIO):
1688  DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT ('1'), (A); END;
1689  PUT FILE(SYSPRINT) SKIP LIST (' UNDER 4 [ ]'), CALL PRGWAX2(11);
1690  PUT FILE(SYSPRINT) SKIP LIST (' 4 - 10 [ ]'), CALL PRGWAX2(12);
1691  PUT FILE(SYSPRINT) SKIP LIST (' 10 - 15 [ ]'), CALL PRGWAX2(13);
1692  PUT FILE(SYSPRINT) SKIP LIST (' 15 - 20 [ ]'), CALL PRGWAX2(4);
1693  PUT FILE(SYSPRINT) SKIP LIST (' 20 - 22 [ ]'), CALL PRGWAX2(5);
1694  PUT FILE(SYSPRINT) SKIP LIST (' 22 - 24 [ ]'), CALL PRGWAX2(6);
1695  PUT FILE(SYSPRINT) SKIP LIST (' 24 - 26 [ ]'), CALL PRGWAX2(7);
1696  PUT FILE(SYSPRINT) SKIP LIST (' 26 - 28 [ ]'), CALL PRGWAX2(8);
1697  PUT FILE(SYSPRINT) SKIP LIST (' 28 - 30 [ ]'), CALL PRGWAX2(9);
1698  PUT FILE(SYSPRINT) SKIP LIST (' 30 - 32 [ ]'), CALL PRGWAX2(10);
1699  PUT FILE(SYSPRINT) SKIP LIST (' 32 - 34 [ ]'), CALL PRGWAX2(11);
1700  PUT FILE(SYSPRINT) SKIP LIST (' 34 - 36 [ ]'), CALL PRGWAX2(12);
1701  PUT FILE(SYSPRINT) SKIP LIST (' 36 - 38 [ ]'), CALL PRGWAX2(13);
1702  PUT FILE(SYSPRINT) SKIP LIST (' 38 - 40 [ ]'), CALL PRGWAX2(14);
1703  PUT FILE(SYSPRINT) SKIP LIST (' 40 - 45 [ ]'), CALL PRGWAX2(15);
1704  PUT FILE(SYSPRINT) SKIP LIST (' 45 - 50 [ ]'), CALL PRGWAX2(16);
1705  PUT FILE(SYSPRINT) SKIP LIST (' 50 - 55 [ ]'), CALL PRGWAX2(17);
1706  PUT FILE(SYSPRINT) SKIP LIST (' 55 - 60 [ ]'), CALL PRGWAX2(18);
1707  PUT FILE(SYSPRINT) SKIP LIST (' 60 - 65 [ ]'), CALL PRGWAX2(19);
1708  PUT FILE(SYSPRINT) SKIP LIST (' 65 - 70 [ ]'), CALL PRGWAX2(20);
1709  PUT FILE(SYSPRINT) SKIP LIST (' 70 - 75 [ ]'), CALL PRGWAX2(21);
1710  PUT FILE(SYSPRINT) SKIP LIST (' 75 - 80 [ ]'), CALL PRGWAX2(22);
1711  PUT FILE(SYSPRINT) SKIP LIST (' 80 - 85 [ ]'), CALL PRGWAX2(23);
1712  PUT FILE(SYSPRINT) SKIP LIST (' 85 - 90 [ ]'), CALL PRGWAX2(24);
1713  PUT FILE(SYSPRINT) SKIP LIST (' 90 - 95 [ ]'), CALL PRGWAX2(25);
1714  PUT FILE(SYSPRINT) SKIP LIST (' OVER 95 [ ]'), CALL PRGWAX2(26);
1715  PUT FILE(SYSPRINT) SKIP LIST (' TOTAL [ ]'), CALL PRGWAX2(27);
1716  PUT FILE(SYSPRINT) SKIP LIST (' VEHICLES [ ]'), CALL PRGWAX2(27);
1717  PUT FILE(SYSPRINT) SKIP LIST (' MEAN-GROSS [ ]');
1718  DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT ('1'), (A); END;
1719  PUT FILE(SYSPRINT) SKIP LIST (' WEIGHT [ ]');
1720  DO I=16 TO 29; PUT FILE(SYSPRINT) EDIT (SOGWAX(1, 1)), (1, F(5, 1),
       A); END;
1721  PUT FILE(SYSPRINT) EDIT (' --- [ ]'), (A);
1722  PUT FILE(SYSPRINT) SKIP LIST (' STANDARD [ ]');
1723  DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT ('1'), (A); END;
1724  PUT FILE(SYSPRINT) SKIP LIST (' DEVIATION [ ]');
1725  DO I=16 TO 29; PUT FILE(SYSPRINT) EDIT (SOGWAX(2, 1)), (1, F(5, 1),
       A); END;
1726  PUT FILE(SYSPRINT) EDIT (' --- [ ]'), (A);
1727  PUT FILE(SYSPRINT) SKIP(PO);
1728  DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT (''), (A); END;
1729  END GWAXC;
1730  PRGWAX1 : PROCEDURE (CA);
VEHDATA: PROCEDURE_OPTIONS (MAIN):

1773  DCL CA FIXED BINARY (31);
1774  DO I=1 TO 15;
1775  PUT FILE(SYSPRINT) EDIT (GOWYSAX(I,CA),* | ') (F(5,0),A); END;
1776  END PRGMAI;
1777  PRGMA2: PROCEDURE (CA);
1778  DCL CA FIXED BINARY (31);
1779  DO I=16 TO 30;
1780  PUT FILE(SYSPRINT) EDIT (GOWYSAX(I,CA),* | ') (F(5,0),A); END;
1781  END PRGMA2;
1782  OSVSAXC: PROCEDURE;
1783  DO I=1 TO 29;
1784  N = OSVSAX(I,1) * (OSVSAX(I,1) - 1);
1785  IF N=0 THEN DO; SDOSAX(I+1) = 0; GO TO G1; END;
1786  DWT = (OSVSAX(I,1) * SDOSAX(I+1) - SDOSAX(I,1) ** 2) / N;
1787  IF DWT<0 THEN DO; SDOSAX(I+1) = -1; GO TO G1; END;
1788  SDOSAX(I+1) = SQRT(DWT);
1789  G1: IF OSVSAX(I,1) = 0 THEN DO; SDOSAX(I+1) = 0; GO TO G2; END;
1790  SDOSAX(I,1) = SDOSAX(I+1) / OSVSAX(I,1);
1791  OSVSAX(30,1) = OSVSAX(30,1) + OSVSAX(I,1);
1792  G2: END;
1793  PUT FILE(SYSPRINT) PAGE LINE(4) EDIT ('OPERATING SPEED VERSUS AX ',
1794  'LE PLACEMENT', 'TANDEM SPAC IS 40 INCHES OR LESS', 'AAHOS',
1795  'CATEGORIES', '1,2 AND 3 INDICATE SINGLE, TANDEM AND TRITA',
1796  'NDEM AXLES', (COLUMN(40), A, A, SKIP(2), COLUMN(39), A, COLUMN(40),
1797  A, A, SKIP(2), COLUMN(31), A, A);
1798  PUT FILE(SYSPRINT) SKIP(5) EDIT ('AXLE PLACEMENT', (COLUMN(53), A);
1799  PUT FILE(SYSPRINT) SKIP EDIT ('OPERATING UNDER', 'SPEED', | 2,
1800  # TONS', | (NPH) | 110 110 111 120 1111 1210 |,
1801  '1120 1300 1111 1210 1120 1112 13100 11300 |,' |' |' |' |' |' |
1802  '2200') (A, A, A, A, A, A, A);
1803  PUT FILE(SYSPRINT) SKIP(0) EDIT ('-------------------------------------------',
1804  '-----------------------------', (A, A, A);
1805  PUT FILE(SYSPRINT) SKIP EDIT ('UNDER 20', | (A, A, A);
1806  PUT FILE(SYSPRINT) SKIP EDIT (20-40 | (A, A, A, A, A, A, A);
1807  PUT FILE(SYSPRINT) SKIP EDIT (40-50 | (A, A, A, A, A, A, A);
1808  PUT FILE(SYSPRINT) SKIP EDIT (50-55 | (A, A, A, A, A, A, A);
1809  PUT FILE(SYSPRINT) SKIP EDIT (55-60 | (A, A, A, A, A, A, A);
1810  PUT FILE(SYSPRINT) SKIP EDIT (60-65 | (A, A, A, A, A, A, A);
1811  PUT FILE(SYSPRINT) SKIP EDIT (65-70 | (A, A, A, A, A, A, A);
1812  PUT FILE(SYSPRINT) SKIP EDIT (70-80 | (A, A, A, A, A, A, A);
1813  PUT FILE(SYSPRINT) SKIP EDIT (80-90 | (A, A, A, A, A, A, A);
1814  PUT FILE(SYSPRINT) SKIP EDIT (OVER 90 | (A, A, A, A, A, A, A);
1815  PUT FILE(SYSPRINT) SKIP EDIT (TOTAL | (A, A, A, A, A, A, A);
1825  DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT(SDOSAX(2), | (F(5,1); A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, A); END;
1826  DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT(SDOSAX(2), | (F(5,1); A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, A); END;
--- VEHDATA : PROCEDURE OPTIONS (MAIN);

PROCEDURE OPTIONS (MAIN);

1854 PUT FILE (SYSPRINT) SKIP (1) EDIT ('AXLE PLACEMENT') (COLUMN(53), A);
1855 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1856 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, SKIP, A, X(98), A, SKIP, A, A, A);
1857 PUT FILE (SYSPRINT) SKIP (1) EDIT ('TOTAL') (A, A, A);
1858 PUT FILE (SYSPRINT) SKIP (1) EDIT ('TOTAL') (A);
1859 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1860 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1861 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1862 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1863 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1864 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1865 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1866 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1867 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1868 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1869 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1870 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1871 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1872 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1873 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1874 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1875 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1876 PUT FILE (SYSPRINT) SKIP EDIT ('SPEED') (MPH)
1877 PUT FILE (SYSPRINT) SKIP EDIT ('TOTAL') (A);
1878 DO I = 1 TO 15; PUT FILE (SYSPRINT) EDIT (* ) (A); END;
1879 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1880 PUT FILE (SYSPRINT) SKIP EDIT ('MEAN') (A);
1881 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1882 PUT FILE (SYSPRINT) SKIP EDIT ('STANDARD') (A);
1883 PUT FILE (SYSPRINT) SKIP EDIT ('VEHICLES') (A, A, A);
1884 PUT FILE (SYSPRINT) SKIP EDIT ('DEVIATION') (A);
1885 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1886 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1887 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1888 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1889 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1890 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1891 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1892 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1893 PUT FILE (SYSPRINT) EDIT ('VEHICLES') (A, A, A);
1894 PUT FILE (SYSPRINT) EDIT ('DEVIATION') (A);
1895 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1896 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1897 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1898 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1899 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1900 PUT FILE (SYSPRINT) EDIT ('SPEED') (MPH)
1901 PUT FILE (SYSPRINT) SKIP (10) EDIT ('------') (A, A, A);
1902 END OSVSAX1;
1903 PROSAX1 : PROCEDURE (CA);
1904 DCL CA FIXED BINARY (31);
1905 DO I = 1 TO 15;
1906 PUT FILE (SYSPRINT) EDIT (OSVSAX1(IC), ' | | ) (F(5, 1), A);
1907 END PROSAX1;
1908 PROSAX2 : PROCEDURE (CA);
1909 DCL CA FIXED BINARY (31);
1910 DO I = 16 TO 30;
1911 PUT FILE (SYSPRINT) EDIT (OSVSAX2(IC), ' | | ) (F(5, 1), A);
1912 END PROSAX2;
1913 PRLOAX1 : PROCEDURE (CA);
1914 DECLARE CA FIXED BINARY (31);
1915 DO I = 16 TO 30;
1916 PUT FILE (SYSPRINT) EDIT (AXLDYSAX1(IC), ' | | ) (F(5, 1), A);
1917 END PRLOAX1;
1918 PRLOAX2 : PROCEDURE (CA);
1919 DECLARE CA FIXED BINARY (31);
1920 DO I = 16 TO 30;
1924  PUT FILE(SYSPRINT) EDIT (AXLDVSAXAA(I,CA)), (F15.0), A); END;
1926  END PRLDAX2;
1927  LOAXCA : PROCEDURE;
1928  DO I=1 TO 29; N = AXLDVSAXAA(I,20) + (AXLDVSAXAA(I,20) - 11);
1930  IF N=0 THEN DO; SDLDAXAA(I,2,1)=0; GO TO A1; END;
1932  CALL PRLDAX(1); N = SDLDAXAA(I,2,1); GO TO A1;
1934  IF N < 0 THEN DO; SDLDAXAA(I,2,1) = 1; GO TO A1; END;
1935  CALL PRLDAX(2); SDLDAXAA(I,2,1) = SQRT(DWT);
1936  A1: IF AXLDVSAXAA(I,20) = 0 THEN DO; SDLDAXAA(I,1,1)=1; GO TO A2; END;
1937  SDLDAXAA(I,1,1) = A1; / AXLDVSAXAA(I,20);
1938  SDLDAXAA(30,20) = AXLDVSAXAA(30,20) + AXLDVSAXAA(1,20);
1939  AXLDVSAXAA(30,21) = AXLDVSAXAA(30,21) + AXLDVSAXAA(21); END;
1941  END;
1942  PUT FILE(SYSPRINT) PAGE LINE:(') EDIT ("AXLE LOAD VERSUS AXLE PLAC",
1943  "PLACED ON") (COLUMN(15),A,A);
1944  PUT FILE(SYSPRINT). SKIP(2) EDIT ("TANDEM SPACING IS 40 INCHES OR ",
1945  "LESS") (COLUMN(14),A,A);
1946  PUT FILE(SYSPRINT). SKIP EDIT ("1, 2 AND 3 INDICATE SINGLE, BITE CAN,
1947  "DEN AND TRITANDEM AXLES"); (COLUMN(31),A,A);
1948  PUT FILE(SYSPRINT). SKIP(5) EDIT ("AXLE UNDER", "AXLE PLACEM",
1949  "ENT") (AXL X(36),A,A);
1950  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1951  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1952  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1953  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1954  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1955  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1956  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1957  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1958  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1959  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1960  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1961  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1962  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1963  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1964  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1965  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1966  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1967  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1968  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1969  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1970  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1971  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1972  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1973  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1974  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1975  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1976  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1977  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1978  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1979  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1980  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1981  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1982  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1983  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1984  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1985  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1986  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1987  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1988  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1989  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1990  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1991  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1992  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1993  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1994  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1995  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1996  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1997  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1998  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
1999  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2000  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2001  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2002  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2003  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2004  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2005  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2006  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
2007  PUT FILE(SYSPRINT). SKIP EDIT ("LOAD") (2 TONS) (A);
DO 1=1 TO 15) (A,15 (F(5,1),A));
  PUT FILE(SYSPRINT). SKIP EDIT ("STANDARD",( ) ) ! DO 1=1 TO
2009  15) (A);
  PUT FILE(SYSPRINT). SKIIP EDIT ("DEVIATION",( ) )
2010  ! DO 1=1 TO 29) (A);
  PUT FILE(SYSPRINT). PAGE LINE(6) EDIT ("AXLE LOAD VERSUS AXLE
2011  PLACEMENT",( ) ) ! COLUMN(30),A,A);
  PUT FILE(SYSPRINT). SKIIP EDIT ("TANDEM SPACING IS 40 INCHES OR
2012  \LESS",( ) ) ! COLUMN(41),A,A);
  PUT FILE(SYSPRINT). SKIIP EDIT ("AASHO CATEGORIES",( ) ) ! COLUMN(49),A);
  PUT FILE(SYSPRINT). SKIIP EDIT ("1 TO 3 INDICATE SINGLE UNIT,
2013  \DEM AND TRITANDEM AXLES",( ) ) ! COLUMN(31),A,A);
  PUT FILE(SYSPRINT). SKIP(51) EDIT ("AXLE LOAD",( ) ) ! AXLE
2014  \PLACEMENTS")
  PUT FILE(SYSPRINT). SKIP(51) EDIT ("LOAD",( ) ) ! (A,X(42),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("UNDE",1,( ) ) ! (A X(98),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("3",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("5",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("7",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("9",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("11",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("13",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("15",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("17",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("19",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("21",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("23",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("25",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("27",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("29",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("31",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("33",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("35",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("OVER",1,( ) ) ! (A,(A),A);
  PUT FILE(SYSPRINT). SKIP EDIT ("TOTAL",1,( ) ) ! (A,
2055  A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("TOTAL",1,( ) ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("Vehicles",1,( ) ) ! (A,
2088  A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("WEIGHT",1,( ) ) ! (A,
2089  1,1,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("WEIGHT",1,( ) ) ! (A,
2092  1,1,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("DEVIATION",1,( ) ) ! (A,
2095  1,1,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("DEVIATION",1,( ) ) ! (A,
2098  1,1,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("STANDARD",1,( ) ) ! (A,
2101  1,1,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
  PUT FILE(SYSPRINT). SKIP EDIT ("STANDARD",1,( ) ) ! (A,
2104  1,1,A,1,1,A);
  PUT FILE(SYSPRINT). EDIT ("" ) ! (A,A,1,1,A);
PROCEDURE (CA); 
DECLARE CA FIXED BINARY(31); 
DO I=1 TO 15; 
CALL PRLDAX3; 
END PRLDAX3;

PROCEDURE (CA); 
DECLARE CA FIXED BINARY(31); 
DO I=16 TO 30; 
CALL PRLDAX4; 
END PRLDAX4;

LDAXCK : PROCEDURE; 
DO I=1 TO 29; N = AXLDVSAX-1;201 /AXLDVSAX-1;201 = 11; 
IF N=0 THEN DO; SDLAX 2;1 = 0; GO TO A1; END; 
DWT = AXLDVSAX-1;201 *SDLAX 2;1 - SDLAX 1;1 *62/ N; 
IF DWT < 0 THEN DO; SDLAX 2;1 = 0; GO TO A1; END; 
STEP SDLAX 2;1 SORTCS; 
A1: IF AXLDVSAX-1;201 THEN DO; SDLAX 1;1 = 0; GO TO A2; END; 
SDLAX 1;1 = SDLAX 1;1 / AXLDVSAX-1;201; 
AXLDVSAX-3;201 = AXLDVSAX-3;201 + AXLDVSAX-1;201; 
END; 
END; 

PUT FILE(SYSPRINT) PAGE LINE(6) EDIT(*AXLE LOAD VERSUS AXLE PLAC*, 
*EMENT) (COLUMN(43),A,A); 
PUT FILE(SYSPRINT) SKIP EDIT(*KENTUCKY CATEGORIES*), 
(COLUMN(47),A); 
PUT FILE(SYSPRINT) SKIP EDIT(*1,2 AND 3 INDICATE SINGLE,BITAN*, 
*EN AND TRITANDEM AXLES*) (COLUMN(31),A,A); 
PUT FILE(SYSPRINT) SKIP(5) EDIT(*... AXLE... UNDER*,*AXLE PLACEN*, 
*ENT*) (A,X(36),A,A); 
PUT FILE(SYSPRINT) SKIP EDIT(* LOAD 12 TONS*) (A,A); 
PUT FILE(SYSPRINT) SKIP EDIT(* (KIPS) | 110 110 111 *, 
*120 1111 1210 1210 1300 1111 1210 11210 11120*, 
* 1300 1300 1220*) (A,A,A); 
PUT FILE(SYSPRINT) SKIP(0) EDIT(*); 

PUT FILE(SYSPRINT) SKIP EDIT(* UNDER 1 */A,A); CALL PRLDAX3(1);
DO I=1 TO 10) (COLUMN(9), A=10, F(7,0), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
DO I=1 TO 20)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* AASHO | MEAN | * |
DO I=1 TO 10)) (COLUMN(9), A=10, F(7,1), A);

PUT FILE(SYSPRINT) SKIP EDIT (* MAXLOADHY(I), * 1)
VEHDATA : PROCEDURE OPTIONS (MAIN);

2319      PUT FILE(SYSPRINT) SKIP EDIT ('*') EAL **,(MINLOADASHO(I),*')
  DO =11 TO 201) (COLUMN(9),A,10 (F(7,1),A));
2320      PUT FILE(SYSPRINT) SKIP EDIT ('*') AASHO | MEAN **,(* | *'
  DO =1 TO 10) (A);
2321      PUT FILE(SYSPRINT) SKIP EDIT ('*') EAL **,( MEANAASHO(I),*'
  DO =11 TO 201) (COLUMN(9),A,10 (F(7,1),A));
2322      PUT FILE(SYSPRINT) SKIP EDIT ('*') STANDARD **,(' | *'
  DO =1 TO 10) (COLUMN(9),11 A);
2323      PUT FILE(SYSPRINT) SKIP EDIT ('*') DEVIATION **, (SDAASHO(I),*
  DO =11 TO 201) (COLUMN(9),A,10 (F(7,1),A));
2324      PUT FILE(SYSPRINT) SKIP EDIT ('*') TOTAL **,(' | *'
  DO =1 TO 10) (COLUMN(9),11 A);
2325      PUT FILE(SYSPRINT) SKIP EDIT ('*') VEHICLES **,(AXLDSAXA(A.21),*'
  DO =11 TO 201) (COLUMN(9),A,10 (F(7,0),A));
2326      PUT FILE(SYSPRINT) SKIP EDIT ('*') ___________ ** DO =1 TO 10)) (A);
2327      PUT FILE(SYSPRINT) PAGE;
2328      PUT FILE(SYSPRINT) SKIP('*) EDIT (* | '*AXLE PLACEMENT*)
2329      (COLUMN(20),A.X1260),A);
2330      PUT FILE(SYSPRINT) SKIP EDIT ('*') CATEGORIES **,* | *'
  DO 122100 112200 121200 13200,'*
  '0 123000 131100 111300 111300 OVER 6*)
2331      (COLUMN(120),A,A);
2332      PUT FILE(SYSPRINT) SKIP(') EDIT ('* | '* DO =1 TO 101)) (A);
2333      PUT FILE(SYSPRINT) SKIP EDIT ('*') MAXIMUM **,'(* | *'
  DO =1 TO 9 ) (COLUMN(9),A,10 A);
2334      PUT FILE(SYSPRINT) SKIP EDIT ('*') MINIMUM **,'(* | *'
  DO =1 TO 9 ) (COLUMN(9),A,10 A);
2335      PUT FILE(SYSPRINT) SKIP EDIT ('*') MAXIMUM **,'(* | *'
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,1),A));
2336      PUT FILE(SYSPRINT) SKIP EDIT ('*') KENTUCKY | MEAN **,* | *'
  DO =1 TO 9 ) (A);
2337      PUT FILE(SYSPRINT) SKIP EDIT ('*') EAL **,(MEANLOADKY(I),*'
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,1),A));
2338      PUT FILE(SYSPRINT) SKIP EDIT ('*') STANDARD **,* | *'
  DO =1 TO 9 ) (COLUMN(9),11 A);
2339      PUT FILE(SYSPRINT) SKIP EDIT ('*') DEVIATION **,(SDKY(I),*
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,1),A));
2340      PUT FILE(SYSPRINT) SKIP EDIT ('*') TOTAL **,* | *'
  DO =1 TO 9 ) (COLUMN(9),11 A);
2341      PUT FILE(SYSPRINT) SKIP EDIT ('*') VEHICLES **,(AXLDSAXA(I,21),*'
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,0),A));
2342      PUT FILE(SYSPRINT) SKIP(') EDIT ('* | '* DO =1 TO 101)) (A);
2343      PUT FILE(SYSPRINT) SKIP EDIT ('*') MAXIMUM **,'(* | *'
  DO =1 TO 9 ) (COLUMN(9),A,10 A);
2344      PUT FILE(SYSPRINT) SKIP EDIT ('*') EAL **,(MAXLOADASHO(I),*'
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,1),A));
2345      PUT FILE(SYSPRINT) SKIP EDIT('*') MINIMUM **,'(* | *'
  DO =1 TO 9 ) (COLUMN(9),A,10 A);
2346      PUT FILE(SYSPRINT) SKIP EDIT ('*') EAL **,(MINLOADASHO(I),*'
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,1),A));
2347      PUT FILE(SYSPRINT) SKIP EDIT ('*') AASHO | MEAN **,* | *'
  DO =1 TO 10) (A);
2348      PUT FILE(SYSPRINT) SKIP EDIT ('*') EAL **,( MEANAASHO(I),*'
  DO =1 TO 291) (COLUMN(9),A,10 (F(7,1),A));
2349      PUT FILE(SYSPRINT) SKIP EDIT ('*') STANDARD **,* | *'

52
DO I=1 TO 9) (COLUMN(9),11 A);  
PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION | ',ISDAASHO(I),', I);  
DO I=21 TO 29) (COLUMN(9),A,10 (F(7,1),A));  
PUT FILE(SYSPRINT) SKIP EDIT ('| TOTAL | ');  
DO I=1 TO 9) (COLUMN(9),11 A);  
PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES | ',AXLOVSAXAA(1,21),', |);  
DO I=21 TO 29) (COLUMN(9),A,10 (F(7,0),A));  
PUT FILE(SYSPRINT) SKIP(O) EDIT ('|_I DO I=1 TO 101)) (A);  
END EWLCE;  
DONE : END VEHDATA;