B. E. KING

COMMISSIONER OF HIGHWAYS

COMMONWEALTH OF KENTUCKY
DEPARTMENT OF HIGHWAYS
FRANKFORT, KENTUCKY 40601

ADDRESS REPLY TO
DEPARTMENT OF HIGHWAYS
DIVISION OF RESEARCH
533 SOUTH LIMESTONE STREET
LEXINGTON, KENTUCKY 40508
TELEPHONE 606-254-4475

June 27, 1971

H-2-27

MEMORANDUM TO: J. R. Harbison, State Highway Engineer
Chairman, Research Committee

SUBJECT: Research Report, "Development of an Electronic Means of Weighing Vehicles in Motion"; KYHPR - 61 - 27; HPR - 1 (6), Part II

Despite overwhelming hardware failures which beset us in the development of an automatic, in-stream, vehicle weighing system -- which we are now convinced we must abandon -- significant measures of success were achieved. In other words, we have created an automaton which almost works. The decision to abandon the prototype installation arose from pilot operations and proof testing. The basic defect is in the weighing platform in the pavement. Unfortunately, it is a design defect.

Tie rods anchoring the platform in the pit induce a purposeful preload on the load-sensing elements. These tie rods change the preload as the temperature fluctuates. Thus, the balance or null point drifts. The noticeable effect was a triggering of the counting and weighing circuits when there was no live load on the platform. Since this load was not transient -- but sustained -- the circuitry "locked in" on the excess preload.

The preload and tie rods were intended to keep the platform in firm bearing on the load-sensing units and to eliminate resonances and friction. Conceivably, it would be possible to control the temperature in the pit, but other factors were equally dissuasive.

The pit structure extends almost four feet below pavement elevation. Access is made by removing the top plates. Whereas walk-in pits were constructed in the entrance ramp to weighing station on I 64, near Shelbyville (Westward side) and on a farm road at the University of Kentucky, it seemed unnecessary to require this feature in roadway installations. In fact, we visualized a "lift-out" platform which could be replaced by a "dummy" if or when repairs were needed. We did not achieve the "lift-out" simplicity.

In the recent past, opportunities to build automatic weighing systems into an Ohio River bridge were forsaken because there was no practical way to flt the platform into the deck system. Consideration was given to incorporating the device into a pit or cavity in the abutment. A later alternative considered was to build the pit and platform completely remote from the bridge -- in a ramp section on an earth embankment. Fortunately, our suspicions regarding the reliability of the pilot installation prevented us from advancing any of the aforesaid plans to the final design stage.

The developmental research on this project began in 1960 and was originally programmed by the Division of Planning [HPS - 1 (22)]. Responsibility transferred to the Division of Research with HPS - HPR - 1 (25), FY 1963-1964. The project was contracted to the University of Kentucky Research Foundation until June 30, 1969. In December 1969, the Research Division was authorized to begin a pilot period of operation. The report submitted herewith pertains only to this final phase -- and our summary evaluation of the system from an operational standpoint. Approximately $198,000 will have

been expended in sustaining the project.

Whereas an early decision was made to adopt the so-called "broken-back bridge" platform in order to achieve a triangular form of output wave as a wheel passed over the platform, others have developed a weighing platform which can be recessed into a pavement (requires only a three-inch inset). The wave form is trapezoidal and would not directly couple with the digitizing system we have. We understand that matching instrument packages will be available soon. This was a persuasive factor in our decision to discontinue this project.

Respectfully submitted,

Jas. H. Havens
Director of Research

Enclosure

cc's: Research Committee
Assistant State Highway Engineer, Research and Development
Assistant State Highway Engineer, Planning and Programming
Assistant State Highway Engineer, Pre-Construction
Assistant State Highway Engineer, Construction
Assistant State Highway Engineer, Operations
Assistant Pre-Construction Engineer
Assistant Operations Engineer
Executive Director, Office of Computer Services
Executive Director, Office of Equipment and 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 Traffic
Director, Division of Planning
Director, Division of Right of Way
Director, Division of Roadside Development
Director, Division of Rural Roads
Division Engineer, Federal Highway Administration
Chairman, Department of Civil Engineering, University of Kentucky
Associate Dean for Continuing Education, College of Engineering, University of Kentucky
All District Engineers

Research Report
307

DEVELOPMENT OF AN ELECTRONIC MEANS OF WEIGHING VEHICLES IN MOTION

FINAL REPORT
KYHPR - 61 - 27, HPR - 1 (6), Part II

by

Ben W. Carr, Jr.
Research Engineer Assistant

and

Rolands L. Rizenbergs
Chief Research Engineer

Division of Research
DEPARTMENT OF HIGHWAYS
Commonwealth of Kentucky

in cooperation with the
U.S. Department of Transportation
Federal Highway Administration

April 1971

ABSTRACT

DEVELOPMENT OF AN ELECTRONIC MEANS OF WEIGHING VEHICLES IN MOTION

An in-stream weighing platform was designed and installed in the eastbound lane of I 64 and 75 near Lexington, Kentucky. The broken-bridge scale platform was designed with the outer edges of the two sections supported on hinges and contiguous edges supported by two 20,000-pound capacity load cells. The assembled scale measures 4' - 6'' x 10' - ½'' with a total weight of about 2,000 pounds.

The electronics developed for the system included digitizing circuitry which processed the load cell signals and recorded the data on digital magnetic tape. Computer processing of the field data produced tabular information on vehicle speed, axle spacing, number of axles, vehicle classification, time of day, and weight for each vehicle, as well as voluminous statistical data such as average daily traffic and equivalent axleloads.

Conceptually, the system was good, but numerous electronic and mechanical problems compounded to render the present system inoperative. Future dynamic weighing system designs should consider portable, lightweight scales and electronic instrumentation suitable for mounting in a vehicle, thus providing a flexible data-gathering system that will be more readily maintainable. Immediate data output in the field would be highly desirable.

TABLE OF CONTENTS

INTRODUCTION

BACKGROUND

The forces which pavements must withstand are dynamic and differ from the static weights of the vehicles. Whereas impact factors relating these forces are sometimes used for designing bridges and pavements, the relationship is variable and only approximate. With regard to law enforcement, it is now necessary to require vehicles to be diverted from the traffic stream and to stop in order to be weighed. A more desirable situation would be to be able to check vehicle weights without requiring them to stop or to at least screen them for suspected overloads.

Basic investigations have been conducted to determine the most practical means of measuring and recording dynamic loads produced by vehicles in motion. The study was initiated in 1960 by the Department of Civil Engineering, University of Kentucky, in cooperation with the Kentucky Department of Highways and the Bureau of Public Roads. Its purpose was to determine the optimum mechanical configuration for a scale which would perform the dynamic axle-weighing function in an overall data-gathering system and to construct and furnish a suitable scale with an appropriate transducer system and automatic data recording system (1, 2, 5). As a result of this study a broken-bridge scale was placed in the outside eastbound lane of I 64 and 75, north of Lexington, Kentucky. An automatic data collection system was installed at the scale site in a vandal-proof structure on the right of way. A final report on that phase of the study was submitted by the University of Kentucky Research Foundation in November 1969 (5).

CURRENT PHASE

Following the completion of the final phase of research and development contracted by the University of Kentucky Research Foundation, in behalf of the subject study, the Division of Research planned to operate the installation for a pilot period to verify the in-stream weighing and recording system. A continuation plan was approved by the Bureau of Public Roads in December 1969. Operation of the scales for data collection and data processing was begun, a method of calibration for weight, speed, axle spacing, and other parameters was developed, and the computer programs for producing statistical data from field data were expanded to meet Departmental needs. Those procedures were developed and used to test the ability of the system to meet the original specifications as defined in February 1967 when the purchase order for the electronic data system was issued to Robert Perelman, DGE Instruments, University Heights, Ohio. These specifications (5) were as follows:

1. The vehicle may have from two to ten axles.
2. The vehicle may travel at legal speeds up to 70 miles per hour; therefore, an actual timing up to 80 miles per hour would be desirable.
3. Dynamic loads will range from 1,000 to 30,000 pounds per axle.
4. The dynamic measurement of load 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 five 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 ½ 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 the load of two axles on the scale simultaneously. Therefore, under this condition, the trace will not return to zero between the 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.

This report covers the pilot-operation period.

RESEARCH EFFORTS

INITIAL EFFORTS - FY 70

After assuming responsibility for the study following the completion of the research and development phase by the University of Kentucky Research Foundation, the Division of Research began a trial period of data collection. Problems with the system became immediately apparent. The signal conditioner had an inherent voltage drift which tended to either mask or diminish the input signal to the digitizing circuitry. The digital tape recorder, which was used to store the field data, was malfunctioning and was returned to the manufacturer for repairs and updating modifications.

Several modifications were made to the original signal conditioning unit. The power supply was redesigned to produce a more stable output voltage. The low pass filter was modified to extract a more ideal output signal. Temperature insensitive components were used throughout the unit. Results of these changes were still not satisfactory. Instead of completely redesigning the unit, two commercially ava lable, highly stable signal conditioning units were purchased and installed in t e system. An active low pass filter and operational amplifiers were used to extract a nearly ideal output signal. Upon its return from the manufacturer, the digital tape recorder required further modifications to re-adapt it for use with the output circuitry of the system. These modifications were required to make the logic levels of the tape recorder and the digitizing circuitry compatible.

Component failures and replacements were common. Some of the components used in the original construction of the system were of poor quality, others had marginal ratings, others were the wrong components to use altogether. Plastic transistors used to drive the indicator lights had to be replaced en masse by transistors with higher power ratings. Throughout the system, several transistors, integrated circuits, and capacitors were replaced. Precision resistors were used to replace common resistors in application where resistance variation had to be held to a minimum.

Loose circuit cards and poor electrical connections between the cards and their connectors were a constant problem. The cards were of such poor construction that they seldom matched with their connectors and had to be forced into a correct position. The printed circuits were not tinned, i.e. covered with solder, and the surface layer of the copper paths quickly oxidized. Each card had to be periodically removed, and the copper paths had to be manually cleaned to assure a proper electrical connection between the card and its connector. Deterioration of the printed circuits, due to the oxidation of the untinned copper, was a continuing source of open circuits and high-resistance current paths.

Throughout the period of the pilot study, the recurring breakdowns of the digitizing circuitry were often traced back to the above mentioned problems with components and circuit cards. These problems were only temporarily alleviated by component replacements and modifications to existing circuit cards and could have been eliminated only by a complete redesign and proper construction of the entire system.

OTHER PROBLEMS

With the system functioning properly, the scales were statically calibrated using a vehicle of known weight. Preparations were made to check accuracy in determining the speed, axle spacing, and weight of several test vehicles. Before these tests were run, a voltage transient in the system damaged one of the two new signal conditioners, destroyed several integrated circuits and other semiconductor components in the digitizing circuitry, and damaged one of the two load cells in the scale platform to the extent that it had to be replaced if data collection were to begin.

ADDITIONAL EFFORTS - FY 71

Due to a lack of sufficient data collection and the numerous problems encountered with the system during FY 70, additional efforts were proposed to verify the system during FY 71. Approval to continue the study was received from the Bureau of Public Roads in August 1970.

To eliminate the cost of purchasing a new load cell, and the inherent delay in delivery, it was decided to replace the above mentioned defective load cell with a previously used, but operative, load cell. This load cell was used in the first phase of this study in the scale installation at the loadometer station near Shelbyville, Kentucky (1). It was also used for approximately one year in the present scales, but had been removed in February 1961 when the other cell in the scales was found to be defective and was replaced (5).

At this time the updated computer programs were stored on magnetic tape, thereby simplifying the processing of field data by virtually eliminating the handling of computer cards.

With the system operational again, serious problems with the scales became apparent. After a vehicle crossed the scales, the output voltage of the signal unit returned to a voltage other than zero. Investigations revealed that hysteresis was caused by insufficient preload. After increasing the preload, a change of temperature in the scale pit caused a drift of signal output. This was determined to be the result of excess preload on the platform.

Another factor entering into the problem of preload adjustment was the oscillations impressed on the platform by a vehicle crossing the scales. Insufficient preload allowed these oscillations to increase in magnitude to such an extent that the digitizing circuitry was triggered, treating the peaks of the oscillations as individual axles.

These problems had been masked by the inherent voltage drift in the original signal conditioning unit. Due to the downtime of the system, they became apparent only after the system was operable over a period of a few days.

While adjusting the preload mechanism, one of the preload rods sheared. New preload rods of a special stainless steel alloy were fabricated and installed in the system. These rods were adjusted to give the best compromise between the effects of hysteresis, oscillations, and output signal drift. Although these adjustments did reduce the above effects, the results were not within acceptable limits. The output signal still had a significant drift which was related to the temperature in the scale pit. Also, the oscillations impressed on the platform by a vehicle crossing the scales significantly distorted the input signal so as to spuriously trigger the digitizing circuitry. Some hysteresis in the platform was still apparent; the platform halves came to rest, following the crossing of the scales by a vehicle, in a position which was slightly higher or lower than the position prior thereto. Thus, it became apparent that to eliminate these problems a new design for the weighing platform was needed.

In October 1970, it was decided by the Division of Research to continue trying to make the system operational but to discontinue work on the present system at the end of the present fiscal year. This decision was prompted by the numerous problems encountered with the system and because of the obvious deterioration of the electronic data system with age.

Data collection was begun, but after a few days a malfunction in the digital tape recorder interferred. A minor repair was made and data collection was begun. A faulty voltage regulator in the tape recorder power supply destroyed several components in the recorder's logic network, again preventing data collection.

While major repairs on the digital tape recorder were in process, the scales were again calibrated, using the same procedures as before. Results of this calibration showed that the load cell installed in the scale in July 1970 was now defective. Its output was approximately 40 per cent below the output of the other load cell. Probable cause of this defect was the load cell's length of time in service and the overloads it had sustained while in use. Thus, replacement of this load cell was necessary.

In February 1971, all efforts to make the system operable were halted. With the scales malfunctioning, the recorder damaged, and the circuit boards deteriorating, further work was deemed unproductive.

RESULTS

As an example of the statistical data obtainable with an operational system, the computer print-out for a single day in November 1970 is included in APPENDIX A. The updated computer program for extracting such data is included in APPENDIX B.

CONCLUSIONS

Conceptually, the system is good, but mechanical and electrical difficulties compound to render the system at hand inoperative. The existing scales, however, have basic inherent deficiencies. Due to its size and weight, the platform exhibits hysteresis, preload, and temperature drift problems of such magnitude as to seriously limit the accuracy obtainable with the system.

Insufficient data has been collected to prove that this system meets the original specifications listed in the INTRODUCTION of this report.

Extensive work had been done to obtain the largest possible amount of statistical data output from the field data. The computer programs designed for use with this system, and adaptable to similar systems, would prove invaluable in highway planning and evaluating applications when used to process extensive amounts of field data.

RECOMMENDATIONS

A portable, lightweight scale similar to the one developed at the University of Texas at Austin should be considered in future in-stream weighing platforms. This scale has no problem with hysteresis and temperature drift and requires no preload. Maintenance can be performed in the laboratory, thus reducing the hazards of on-site repairs and maintenance while diverting traffic around the site. The system is portable, and data can be gathered from several selected sites (7).

The electronics package in future designs should be installed in a van-type vehicle to provide portability. Electronic design should include facilities to store field data and simultaneously display desired results such as vehicle speed, axle weights, gross weights, etc.

REFERENCES

1. " Weighing Vehicles in Motion, A Summary Report," Kentucky Department of Highways, April 1964.

2. O.K. Normann and R.C. Hopkins, "Weighing Vehicles in Motion," Bulletin 50, Highway Research Board, 1952.

3. "Weighing Vehicles in Motion, A Progress Report," Kentucky Department of Highways, October 1965.

4. "RODAS (Road Data Acquisition System) Operation and Instruction Manual," Office of Research and Engineering Services, College of Engineering, University of Kentucky, June 1969.

5. "Research Relating to Weighing Vehicles in Motion," Office of Research and Engineering Services, College of Engineering, University of Kentucky, September 1969.

6. S.P. Maggard, "Weighing Vehicles in Motion," Unpublished Thesis, Master of Science in Civil Engineering, University of Kentucky 1957.

7. Lee, C.E. and Al-Rashid, N.I., "A Portable Electronic Scale for Weighing Vehicles in Motion," Center for Highway Research, The University of Texas at Austin, April 1968.

APPENDIX A

Example of Output Data

VEHICLE DATA FOR DAY 112

STATION IDENTIFICATION = 4

NUMBER OF AXLES FOR THIS DAY = 9557

NUMBER OF VEHICLES FOR THIS DAY = 4280

CURRENT LOAD EQUIVALENCY FACTORS (AS OF 3-20-69)

SINGLE AXLES			TANDEM AXLES		
LOAD (KIPS)	KENTUCKY	AASHO	LOAD (KIPS)	KENTUCKY	AASHO
1-3	0	0.0002	2-6		
3-5	0	0.002	6-10		
5-7	0	0.01	10-14		0.01
7-9	0	0.03	14-18		0.05
9-11	1	0.09	18-22		0.12
11-13	2	0.19	22-26		0.26
13-15	4	0.36	26-30		0.50
15-17	8	0.62	30-34		0.86
17-19	16	1.00	34-38		1.38
19-21	32	1.51	38-42		2.08
21-23	64	2.18	42-46		3.00
23-25	128	3.03	46-50		4.17
25-27	256	4.09	50-54		5.63
27-29	512	5.39	54-58		7.41
29-31	1024	6.97	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. IN ADDITION, TWO-AXLE, FOUR TIRED VEHICLES ARE ASSUMED TO CONTRIBUTE 0.0002 EAL'S PER VEHICLE.
SINGLE AXLE, AASHO FACTORS RELATE TO FLEXIBLE PAVEMENTS HAVING A TERMINAL SERVICEABILITY INDEX OF 2.5 AND A STRUCTURAL NUMBER OF 5.

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

0	0			110
0	0	0		111
0	0	0	0	1111
0	00	0		1210
0	0	00	0	1121
0	0	00		1120
0	000			1300

GROSS OPERATING WEIGHT VERSUS OPERATING SPEED

GROSS OPERATING WEIGHT (KIPS)	SPEED (MPH)										
	UNDER 20	20-40	40-50	50-55	55-60	60-65	65-70	70-80	80-90	OVER 90	TOTAL VEHICLES
UNDER 4	0	2	2	11	8	0	0	3	0	1	27
4 - 10	0	1	2	3	2	0	0	2	0	0	10
10 - 15	0	0	0	2	1	0	0	0	0	0	3
15 - 20	0	0	0	1	1	0	0	0	0	0	2
20 - 22	0	0	0	0	0	0	0	0	0	0	0
22 - 24	0	0	0	1	0	0	0	0	0	0	1
24 - 26	0	0	0	0	0	0	0	0	0	0	0
26 - 28	0	0	0	0	0	0	0	0	0	0	0
28 - 30	0	0	0	1	0	0	0	0	0	0	1
30 - 32	0	0	0	0	0	0	0	0	0	0	0
32 - 34	0	0	0	0	0	0	0	0	0	0	0
34 - 36	0	0	0	0	0	0	0	0	0	0	0
36 - 38	0	0	0	0	0	0	0	0	0	0	0
38 - 40	0	0	0	0	0	0	0	0	0	0	0
40 - 45	0	270	530	1347	977	0	0	371	165	25	3685
45 - 50	0	0	0	0	0	0	0	0	0	0	0
50 - 55	0	0	0	0	0	0	0	0	0	0	0
55 - 60	0	0	0	0	0	0	0	0	0	0	0
60 - 65	0	0	0	0	0	0	0	0	0	0	0
65 - 70	0	32	30	92	61	0	0	20	6	2	243
70 - 75	0	0	0	0	0	0	0	0	0	0	0
75 - 80	0	0	0	0	0	0	0	0	0	0	0
80 - 85	0	0	0	0	0	0	0	0	0	0	0
85 - 90	0	25	43	78	31	0	0	13	8	2	200
90 - 95	0	0	0	0	0	0	0	0	0	0	0
OVER 95	0	11	15	43	23	0	0	7	8	1	108
TOTAL VEHICLES	0	341	622	1579	1104	0	0	416	187	31	4280
MEAN GROSS WEIGHT	0.0	51.1	49.5	48.9	47.4	0.0	0.0	47.1	49.8	49.1	----
STANDARD DEVIATION	0.0	17.4	15.8	15.7	13.5	0.0	0.0	13.2	17.5	18.3	----

GROSS OPERATING WEIGHT VERSUS AXLE PLACEMENT
(AASHO CATEGORIES)

TANDEM SPACING IS 40 INCHES OR LESS

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

GROSS OPERATING WEIGHT (KIPS)	UNDER 2 TONS	AXLE PLACEMENT														
	110	110	111	120	1111	1210	1120	1300	11111	12110	11210	11120	13100	11300	12200
UNDER 4	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 - 10	0	6	1	0	3	0	0	0	0	0	0	0	0	0	0
10 - 15	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0
15 - 20	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
20 - 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22 - 24	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
24 - 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26 - 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28 - 30	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
30 - 32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32 - 34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34 - 36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36 - 38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38 - 40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40 - 45	0	3685	0	0	0	0	0	0	0	0	0	0	0	0	0
45 - 50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50 - 55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55 - 60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60 - 65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65 - 60	0	0	241	1	0	0	0	0	1	0	0	0	0	0	0
70 - 75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75 - 80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80 - 85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85 - 90	0	0	0	0	197	1	2	0	0	0	0	0	0	0	0
90 - 95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 95	0	0	0	0	0	0	0	0	91	0	0	2	1	0	2
TOTAL VEHICLES	27	3694	244	1	202	1	2	0	92	0	0	2	1	0	2
MEAN GROSS WEIGHT	2.5	43.9	65.3	66.0	86.1	88.0	88.0	0.0	109.5	0.0	0.0	110.0	110.0	0.0	110.0
STANDARD DEVIATION	0.4	61.3	-1.0	0.0	-1.0	0.0	0.0	0.0	4.3	0.0	0.0	0.0	0.0	0.0	0.0

GROSS OPERATING WEIGHT VERSUS AXLE PLACEMENT (CONTINUED FROM PRECEDING PAGE)

TANDEM SPACING IS 40 INCHES OR LESS
(AASHO CATEGORIES)

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

GROSS OPERATING WEIGHT (KIPS)	AXLE PLACEMENT														TOTAL VEHICLES
	111111	121110	112110	111210	111120	122100	112200	121200	132000	123000	131100	113100	111300	OVER 6	
UNDER 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27
4 - 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
10 - 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
15 - 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
20 - 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22 - 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
24 - 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26 - 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28 - 30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
30 - 32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32 - 34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34 - 36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36 - 38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38 - 40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40 - 45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3685
45 - 50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50 - 55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55 - 60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60 - 65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65 - 70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	243
70 - 75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75 - 80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80 - 85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85 - 90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200
90 - 95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 95	9	1	0	0	1	0	0	0	0	0	0	0	0	1	108
TOTAL VEHICLES	9	1	0	0	1	0	0	0	0	0	0	0	0	1	4280
MEAN GROSS WEIGHT	132.0	132.0	0.0	0.0	132.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	154.0	----
STANDARD DEVIATION	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	----

OPERATING SPEED VERSUS AXLE PLACEMENT

TANDEM SPACING IS 40 INCHES OR LESS
(AASHO CATEGORIES)

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

AXLE PLACEMENT

OPERATING SPEED (MPH)	UNDER 2 TONS	110	110	111	120	1111	1210	1120	1300	11111	12110	11210	11120	13100	11300	12200
UNDER 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20 - 40	2	271	32	0	24	0	1	0	8	0	0	1	0	0	1
40 - 50	2	532	30	0	41	1	1	0	12	0	0	0	1	0	1
50 - 55	11	1349	93	1	81	0	0	0	39	0	0	1	0	0	0
55 - 60	8	980	61	0	32	0	0	0	21	0	0	0	0	0	0
60 - 65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65 - 70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70 - 80	3	372	20	0	14	0	0	0	6	0	0	0	0	0	0
80 - 90	0	165	6	0	8	0	0	0	5	0	0	0	0	0	0
OVER 90	1	25	2	0	2	0	0	0	1	0	0	0	0	0	0
TOTAL VEHICLES	27	3694	244	1	202	1	2	0	92	0	0	2	1	0	2
MEAN SPEED	56.7	55.7	53.8	50.5	52.9	49.6	39.1	0.0	54.8	0.0	0.0	44.4	40.1	0.0	35.7
STANDARD DEVIATION	11.6	448.4	-1.0	0.0	-1.0	0.0	2.9	0.0	11.5	0.0	0.0	9.8	0.0	0.0	7.0

AXLE PLACEMENT

OPERATING SPEED (MPH)	111111	121110	112110	111210	111120	122100	112200	121200	132000	123000	131100	113100	111300	OVER 6	TOTAL VEHICLES
UNDER 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20 - 40	0	1	0	0	0	0	0	0	0	0	0	0	0	0	341
40 - 50	0	0	0	0	1	0	0	0	0	0	0	0	0	0	622
50 - 55	3	0	0	0	0	0	0	0	0	0	0	0	0	1	1579
55 - 60	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1104
60 - 65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65 - 70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70 - 80	1	0	0	0	0	0	0	0	0	0	0	0	0	0	416
80 - 90	3	0	0	0	0	0	0	0	0	0	0	0	0	0	187
OVER 90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31
TOTAL VEHICLES	9	1	0	0	1	0	0	0	0	0	0	0	0	1	4280
MEAN SPEED	67.5	39.0	0.0	0.0	40.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	52.3	----
STANDARD DEVIATION	15.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	----

AXLE LOAD VERSUS AXLE PLACEMENT

TANDEM SPACING IS 40 INCHES OR LESS
(AASHO CATEGORIES)

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

AXLE LOAD (KIPS)	UNDER 2 TONS	AXLE PLACEMENT														
	110	110	111	120	1111	1210	1120	1300	11111	12110	11210	11120	13100	11300	12200
UNDER 1	6	2	2	0	4	0	0	0	0	0	0	0	0	0	0
1 - 3	48	6	3	0	9	0	0	0	2	0	0	0	0	0	0
3 - 5	0	2	0	0	3	0	0	0	0	0	0	0	0	0	0
5 - 7	0	4	2	0	1	0	0	0	0	0	0	0	0	0	0
7 - 9	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
9 - 11	0	0	2	0	3	0	0	0	0	0	0	0	0	0	0
11 - 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13 - 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15 - 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17 - 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19 - 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21 - 23	0	7371	723	1	788	2	4	0	458	0	0	6	2	0	2
23 - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25 - 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27 - 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29 - 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31 - 33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33 - 35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 35	0	0	0	1	0	1	2	0	0	0	0	2	1	0	4
TOTAL AXLES	54	7388	732	2	808	3	6	0	460	0	0	8	3	0	6
TOTAL VEHICLES	27	3694	244	1	202	1	2	0	92	0	0	2	1	0	2
MEAN AXLE WEIGHT	1.3	22.0	21.8	33.0	21.5	29.3	29.3	0.0	21.9	0.0	0.0	27.5	36.7	0.0	36.7
STANDARD DEVIATION	0.3	-1.0	14.2	15.6	-1.0	12.7	11.4	0.0	5.2	0.0	0.0	10.2	25.4	0.0	11.4

AXLE LOAD VERSUS AXLE PLACEMENT (CONTINUED FROM LAST PAGE)

TANDEM SPACING IS 40 INCHES OR LESS
(AASHO CATEGORIES)

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

AXLE LOAD (KIPS)	AXLE PLACEMENT															
	111111	121110	112110	111210	111120	122100	112200	121200	132000	123000	131100	113100	111300	OVER 6	TOTALS
UNDER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14
1 - 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68
3 - 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
5 - 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
7 - 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
9 - 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
11 - 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13 - 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15 - 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17 - 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19 - 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21 - 23	54	4	0	0	4	0	0	0	0	0	0	0	0	7	9426
23 - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25 - 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27 - 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29 - 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31 - 33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33 - 35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 35	0	1	0	0	1	0	0	0	0	0	0	0	0	0	13
TOTAL AXLES	54	5	0	0	5	0	0	0	0	0	0	0	0	7	9541
TOTAL VEHICLES	9	1	0	0	1	0	0	0	0	0	0	0	0	1	4280
MEAN AXLE WEIGHT	22.0	26.4	0.0	0.0	26.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22.0	----
STANDARD DEVIATION	0.0	9.8	0.0	0.0	9.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	----

AXLE LOAD VERSUS AXLE PLACEMENT

TANDEM SPACING IS 40 INCHES TO 120 INCHES
(KENTUCKY CATEGORIES)

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

AXLE LOAD (KIPS)	UNDER 2 TONS	AXLE PLACEMENT														
	110	110	111	120	1111	1210	1120	1300	11111	12110	11210	11120	13100	11300	12200
UNDER 1	6	2	2	0	1	0	0	0	0	0	0	0	0	0	0
1 - 3	48	6	3	0	3	0	2	2	0	0	0	0	0	0	1
3 - 5	0	2	0	0	0	0	1	1	0	0	0	0	0	0	0
5 - 7	0	4	2	0	0	0	1	0	0	0	0	0	0	0	0
7 - 9	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
9 - 11	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0
11 - 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13 - 15	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
15 - 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17 - 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19 - 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21 - 23	0	7371	618	35	56	132	192	21	25	6	0	6	0	8	79
23 - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25 - 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27 - 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29 - 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31 - 33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33 - 35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 35	0	0	0	35	0	66	96	21	0	2	0	2	0	4	157
TOTAL AXLES	54	7388	627	70	60	198	294	46	25	8	0	8	0	12	237
TOTAL VEHICLES	27	3694	209	35	15	66	98	23	5	2	0	2	0	4	79
MEAN AXLE WEIGHT	1.3	22.0	21.7	33.0	20.6	29.3	28.9	40.6	22.0	27.5	0.0	27.5	0.0	36.7	36.5
STANDARD DEVIATION	0.3	0.0	0.0	11.1	5.2	0.0	22.1	24.0	0.0	10.2	0.0	10.2	0.0	21.7	0.0

AXLE LOAD VERSUS AXLE PLACEMENT (CONTINUED FROM LAST PAGE)

TANDEM SPACING IS 40 INCHES TO 120 INCHES
(KENTUCKY CATEGORIES)

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

AXLE LOAD (KIPS)	AXLE PLACEMENT																
	111111	121110	112110	111210	111120	122100	112200	121200	132000	123000	131100	113100	111300	OVER 6	TOTALS
UNDER 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
1 - 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65
3 - 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
5 - 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
7 - 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
9 - 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
11 - 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13 - 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
15 - 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17 - 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19 - 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21 - 23	0	8	0	0	0	0	2	0	2	3	0	0	3	7	8574
23 - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25 - 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27 - 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29 - 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31 - 33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33 - 35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OVER 35	0	2	0	0	0	0	2	0	4	6	0	0	1	0	398
TOTAL AXLES	0	10	0	0	0	0	4	0	6	9	0	0	4	7	9067
TOTAL VEHICLES	0	2	0	0	0	0	1	0	2	3	0	0	1	1	4269
MEAN AXLE WEIGHT	0.0	26.4	0.0	0.0	0.0	0.0	33.0	0.0	44.0	44.0	0.0	0.0	33.0	22.0	----
STANDARD DEVIATION	0.0	9.3	0.0	0.0	0.0	0.0	12.7	0.0	19.7	19.1	0.0	0.0	22.0	0.0	----

EQUIVALENT AXLE LOAD PER VEHICLE

CATEGORIES		UNDER 2 TONS	AXLE PLACEMENT								
		110	110	111	120	1111	1210	1120	1300	11111	12110
KENTUCKY	MAXIMUM EAL	0.0	128.0	192.0	1088.0	256.0	1152.0	1152.0	1088.0	320.0	1216.0
	MINIMUM EAL	0.0	0.0	0.0	1088.0	0.0	1152.0	0.0	0.0	320.0	1216.0
	MEAN EAL	0.0	127.7	189.3	1088.0	238.9	1152.0	1128.5	993.6	320.0	1216.0
	STANDARD DEVIATION	0.0	6.0	22.8	0.0	66.1	0.0	163.4	312.9	0.0	0.0
	TOTAL VEHICLES	27	3694	209	35	15	66	98	23	5	2
AASHO	MAXIMUM EAL	0.0	4.4	6.5	5.2	8.7	7.4	7.4	0.0	10.9	0.0
	MINIMUM EAL	0.0	0.0	0.0	5.2	0.0	7.4	7.4	0.0	6.5	0.0
	MEAN EAL	0.0	4.3	6.5	5.2	8.5	7.4	7.4	0.0	10.9	0.0
	STANDARD DEVIATION	0.0	0.2	0.7	0.0	1.3	0.0	1.0	0.0	0.5	0.0
	TOTAL VEHICLES	27	3694	244	1	202	1	2	0	92	0

CATEGORIES		AXLE PLACEMENT									
		11210	11120	13100	11300	12200	111111	121110	112110	111210	111120
KENTUCKY	MAXIMUM EAL	0.0	1216.0	0.0	1152.0	2112.0	0.0	1280.0	0.0	0.0	0.0
	MINIMUM EAL	0.0	1216.0	0.0	1152.0	1088.0	0.0	1280.0	0.0	0.0	0.0
	MEAN EAL	0.0	1216.0	0.0	1152.0	2099.0	0.0	1280.0	0.0	0.0	0.0
	STANDARD DEVIATION	0.0	0.0	0.0	0.0	115.2	0.0	0.0	0.0	0.0	0.0
	TOTAL VEHICLES	0	2	0	4	79	0	2	0	0	0
AASHO	MAXIMUM EAL	0.0	9.5	13.9	0.0	8.2	13.1	11.7	0.0	0.0	11.7
	MINIMUM EAL	0.0	9.5	13.9	0.0	8.2	13.1	11.7	0.0	0.0	11.7
	MEAN EAL	0.0	9.5	13.9	0.0	8.2	13.1	11.7	0.0	0.0	11.7
	STANDARD DEVIATION	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	TOTAL VEHICLES	0	2	1	0	2	9	1	0	0	1

CATEGORIES		AXLE PLACEMENT								
		122100	112200	121200	132000	123000	131100	113100	11130	OVER 6
KENTUCKY	MAXIMUM EAL	0.0	2176.0	0.0	2112.0	2112.0	0.0	0.0	1216.0	448.0
	MINIMUM EAL	0.0	2176.0	0.0	2112.0	2112.0	0.0	0.0	1216.0	448.0
	MEAN EAL	0.0	2176.0	0.0	2112.0	2112.0	0.0	0.0	1216.0	448.0
	STANDARD DEVIATION	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	TOTAL VEHICLES	0	1	0	2	3	0	0	1	1
AASHO	MAXIMUM EAL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.3
	MINIMUM EAL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.3
	MEAN EAL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.3
	STANDARD DEVIATION	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	TOTAL VEHICLES	0	0	0	0	0	0	0	0	1

APPENDIX B

PL/1 Program for Tabular Output

```
//P27CHART  JOB (1009-51001,,,,A),' BENNY DUKES',MSGLEVEL=1,REGION=268K
//BLAH  EXEC PL1LFCLG,PARM.PL1L='EXTDIC,OPT=1',PARM.LKED='LET'
//PL1L.SYSLIN DD SPACE=(80,(990,990))
//PL1L.SYSUT3 DD SPACE=(80,(990,990))
//PL1L.SYSUT1 DD SPACE=(1024,(500,500))
//PL1L.SYSIN DD *
 VEHDATA : PROCEDURE OPTIONS (MAIN);                                          TBL00010
     OPEN FILE(SYSPRINT) OUTPUT LINESIZE(132);                                TBL00020
     DECLARE                          SLASH CHAR(1);                          TBL00030
 DECLARE                                                                      TBL00040
     (OSVSAX(30,11) INIT((330)0),     GOWVSAX(30,27) INIT((810)0),            TBL00050
     GOWVSOS(1),27) INIT((297)0),     AXLDVSAX(30,21) INIT((630)0),           TBL00060
     AXLDVSAXAA(30,21) INIT((630)0),  AL INIT(0),                             TBL00070
     CA INIT(0),                      CAA INIT(0),                            TBL00080
     CAD INIT(0),                     I INIT(0),                              TBL00090
     DAY INIT(0),                     ID INIT(0),                             TBL00100
     NUMOFAXLES INIT(0),              N INIT(0),                              TBL00110
     AXLENUM INITIAL(0),              RECORD INITIAL(0),                      TBL00120
     DAYHOLD INIT(0),                 IDHOLD INIT(0),                         TBL00130
     NUMBER INIT(0),                  NUMVEH INIT(0),                         TBL00140
     SPD INIT(0),                     W INIT(0))                              TBL00150
                        FIXED BINARY(31) STATIC;                              TBL00160
 DECLARE                                                                      TBL00170
     (STORE(9,3) INIT((27)0),         SDOSAX(3,30) INIT((90)0),               TBL00180
     SDGOWAX(3,30) INIT((90)0),       SDGOWOS(2,10) INIT((20)0),              TBL00190
     SDLDAX(3,29) INIT((87)0),        SDLDAXAA(3,29) INIT((87)0),             TBL00200
     MAXLDAASHO(29) INIT((29)0),      LDAASHO INIT(0),                        TBL00210
     MEANAASHO(29) INIT((29)0),       SDAASHO(29) INIT((29)0),                TBL00220
     MAXLOADKY(29) INIT((29)0),       LOADKY INIT(0),                         TBL00230
     MEANLOADKY(29) INIT((29)0),      SDKY(29) INIT((29)0),                   TBL00240
     LD INIT(0),                      VEHWT INIT(0),                          TBL00250
     DWT INIT(0),                     LOAD INIT(0),                           TBL00260
     WEIGHT INIT(0.0),                SPEED INIT(0.0),                        TBL00270
     TA INIT (0),                     TK INIT(0),                             TBL00280
     SPACING INIT(0.0),                                                       TBL00290
     MINLDAASHO(29) INIT((29)99999),  MINLOADKY(29) INIT((29)99999))          TBL00300
                        FLOAT BINARY(16) STATIC;                              TBL00310
     DECLARE   MAXMINA ENTRY (FIXED BINARY (31)),                             TBL00320
               MAXMINK ENTRY (FIXED BINARY (31)),                             TBL00330
               GOWCHK  ENTRY (FIXED BINARY (31)),                             TBL00340
               SPDCHK  ENTRY (FIXED BINARY (31)),                             TBL00350
               AXLOAD ENTRY (FLOAT BINARY(16), FIXED BINARY (31)),            TBL00360
               ADDAXL ENTRY (FIXED BIN(31),FIXED BIN(31),FLOAT BIN(16)),TBL00370
               ADAXLAA ENTRY(FIXED BIN(31),FIXED BIN(31),FLOAT BIN(16)),TBL00380
               ADDVEH ENTRY(FIXED BIN(31),FIXED BIN(31),FIXED BIN(31)), TBL00390
               TANDAA ENTRY (FLOAT BIN (16)),                                 TBL00400
               ADDVHKY ENTRY (FIXED BINARY (31)),                             TBL00410
               ADDVHAA ENTRY (FIXED BINARY (31));                             TBL00510
 ON ENDFILE(TAPE) BEGIN; CALL SUMMARY; GO TO DONE; END;                       TBL00520
 BEGIN: GET FILE(TAPE) EDIT (WEIGHT,SPEED,SPACING,DAY,AXLENUM,ID,SLASH)       TBL00530
          (F(3,1),F(4,1),F(3,1),F(3,0),X(9),2 F(1,0),A(1));                   TBL00540
     IF ID=0 THEN GO TO BEGIN;                                                TBL00550
     IF DAY=0 THEN GO TO BEGIN;                                               TBL00560
     IF SLASH ¬= '/' THEN DO; A1:PUT FILE(SYSPRINT) SKIP(5) EDIT              TBL00570
          ('++++++++++++++++++++++++++++++ 25 NUMBERS WERE NOT FOUND B',TBL00580
          'ETWEEN SLASHES ++++++++++++++++++++++++++++')(A);A2:GET FILE       TBL00590
          (TAPE) EDIT (SLASH)(A(1)); IF SLASH ¬= '/' THEN GO TO A2; A3:       TBL00600
          GET FILE(TAPE) EDIT (WEIGHT,SPEED,SPACING,DAY,AXLENUM,ID,           TBL00610
          SLASH) (F(3,1),F(4,1),F(3,1),F(3,0),X(9),2 F(1,0),A(1));            TBL00620
          IF SLASH ¬= '/' THEN GO TO A1; IF AXLENUM ¬= 1 THEN GO TO A3;       TBL00630
          GO TO G2; END;                                                      TBL00640
```

```
    IF RECORD = 0 THEN DO; RECORD=1; DAYHOLD=DAY; IDHOLD=ID; GO TO G3; TBL00650
       END;                                                          TBL00660
    IF ID ¬= IDHOLD THEN DO; CALL SUMMARY; IDHOLD = ID; DAYHOLD=DAY;  TBL00670
       GO TO G1; END;                                                TBL00680
    IF DAY ¬= DAYHOLD THEN DO; CALL SUMMARY; DAYHOLD=DAY;GO TO G1; END;TBL00690
G1: IF AXLENUM = 1 THEN                                              TBL00700
G2: DO; IF NUMBER=1 THEN DO; STORE(1,1)=WEIGHT; STORE(1,2)=SPEED;    TBL00710
       STORE(1,3)=SPACING; VEHWT= WEIGHT; GO TO BEGIN; END;NUMVEH =  TBL00720
       NUMVEH + 1; GO TO B1; END;                                    TBL00730
G3: NUMBER=AXLENUM; STORE(NUMBER,1) = WEIGHT; STORE(NUMBER,2)=SPEED; TBL00740
       STORE(NUMBER,3)=SPACING; VEHWT = VEHWT + WEIGHT; NUMOFAXLES = TBL00750
       NUMOFAXLES + 1;; GO TO BEGIN;                                 TBL00760
B1: CALL SPDCHK(SPD); CALL GOWCHK(W);                                TBL00770
    IF NUMBER=2 THEN DO; IF VEHWT <= 4.0 THEN CALL AXLE2L; ELSE CALL TBL00780
       AXLE2H; GO TO G4; END;                                        TBL00790
    IF NUMBER=3 THEN DO; CALL AXLE3; GO TO G4; END;                  TBL00800
    IF NUMBER=4 THEN DO; CALL AXLE4; GO TO G4; END;                  TBL00810
    IF NUMBER=5 THEN DO; CALL AXLE5; GO TO G4; END;                  TBL00820
    IF NUMBER=6 THEN DO; CALL AXLE6; GO TO G4; END;                  TBL00830
    CALL AXLEM6;                                                     TBL00840
G4: VEHWT=0.0;      GO TO G3;                                        TBL00850
SUMMARY : PROCEDURE;                                                 TBL00860
    PUT FILE(SYSPRINT) PAGE LINE(2)  EDIT ((80'*' DO I=1 TO 8),      TBL00870
       ' VEHICLE DATA FOR DAY ',DAYHOLD, (80'*' DO I=1 TO 8)),       TBL00880
       (COLUMN(5),9 A,F(3,0),X(1),8 A);                              TBL00890
    PUT FILE(SYSPRINT) SKIP(2) EDIT ('STATION IDENTIFICATION = ',    TBL00900
       IDHOLD) (COLUMN(53),A,F(1,0));                                TBL00910
    PUT FILE(SYSPRINT) SKIP(2) EDIT ('NUMBER OF AXLES FOR THIS DAY =', TBL00920
       NUMOFAXLES) (COLUMN(48),A,F(5,0));                            TBL00930
    PUT FILE(SYSPRINT) SKIP(2) EDIT ('NUMBER OF VEHICLES FOR THIS DAY',TBL00940
       ' =', NUMVEH) (COLUMN(47),2 A,F(5,0));                        TBL00950
    PUT FILE(SYSPRINT) SKIP(4)  EDIT  (                              TBL00960
       '   CURRENT LOAD EQUIVALENCY FACTORS (AS OF 3-20-69)',        TBL00970
       '         SINGLE AXLES                    TANDEM AXLES',      TBL00980
       ' LOAD    KENTUCKY    AASHO       LOAD    KENTUCKY    AASHO', TBL00990
       '(KIPS)                           (KIPS)',                    TBL01000
       '  1-3          0      0.0002       2-6  ',                  TBL01010
       '  3-5          0      0.002        6-10 ',                  TBL01020
       '  5-7          0      0.01        10-14          0.01',     TBL01030
       '  7-9          0      0.03        14-18          0.05',     TBL01040
       '  9-11         1      0.09        18-22          0.12',     TBL01050
       ' 11-13         2      0.19        22-26          0.26',     TBL01060
       ' 13-15         4      0.36        26-30          0.50',     TBL01070
       ' 15-17         8      0.62        30-34          0.86',     TBL01080
       ' 17-19        16      1.00        34-38          1.38',     TBL01090
       ' 19-21        32      1.51        38-42          2.08',     TBL01100
       ' 21-23        64      2.18        42-46          3.00',     TBL01110
       ' 23-25       128      3.03        46-50          4.17',     TBL01120
       ' 25-27       256      4.09        50-54          5.63',     TBL01130
       ' 27-29       512      5.39        54-58          7.41',     TBL01140
       ' 29-31      1024      6.97        58-62          9.59',     TBL01150
       'NOTE:  KENTUCKY DOES NOT IDENTIFY TANDEM AXLES SEPARATELY ', TBL01160
       '       FOR PURPOSES OF COMPUTATION.',                        TBL01170
       '       THE FACTORS USED BY AASHO RELATE TO TRUCK AXLES.  IN',TBL01180
       '       ADDITION, TWO-AXLE, FOUR TIRED VEHICLES ARE ASSUMED', TBL01190
       '       TO CONTRIBUTE 0.0002 EAL''S PER VEHICLE.',            TBL01200
       '       SINGLE AXLE, AASHO FACTORS RELATE TO FLEXIBLE PAVE-', TBL01210
       '       MENTS HAVING A TERMINAL SERVICEABILITY INDEX OF 2.5', TBL01220
       '       AND A STRUCTURAL NUMBER OF 5.')                       TBL01230
       (COLUMN(38),A,SKIP(2),18(COLUMN(36),A),SKIP(2), 8(COLUMN(36), TBL01240
       A));                                                          TBL01250
```

```
    PUT FILE(SYSPRINT) SKIP(4) EDIT (                                          TBL01260
         '1,2 AND 3 INDICATE SINGLE,BITANDEM AND TRITANDEM AXLES',             TBL01270
         '0    0              110 ',                                           TBL01280
         '0    0    0         111 ',                                           TBL01290
         '0    0    0    0    1111',                                           TBL01300
         '0    00   0         1210',                                           TBL01310
         '0    0    00   0    1121',                                           TBL01320
         '0    0    00        1120',                                           TBL01330
         '0    000            1300')                                           TBL01340
         (COLUMN(39),A,SKIP(2), 7 (COLUMN(53),A));                             TBL01350
    CALL  GWVSOSC (GOWVSOS,SDGOWOS);                                           TBL01360
    CALL GWAXC (GOWVSAX,SDGOWAX);                                              TBL01370
    CALL OSVSAXC (OSVSAX,SDOSAX);                                              TBL01380
    CALL  LDAXCA (AXLDVSAXAA,SDLDAXAA);                                        TBL01390
    CALL LDAXCK (AXLDVSAX,SDLDAX);                                             TBL01400
    CALL EWLC;                                                                 TBL01410
    OSVSAX=0; SDOSAX=0; SDGOWAX=0; SDGOWAX=0; AXLDVSAX=0; AXLDVSAXAA=0;        TBL01420
         NUMOFAXLES=0; NUMVEH=0; SDGOWOS=0; SDLDAX=0; SDLDAXAA=0;              TBL01430
         MAXLDAASHO=0; MEANAASHO=0; SDAASHO=0; MAXLOADKY=0;                    TBL01440
         MINLDAASHO=99999; MINLOADKY=99999; MEANLOADKY=0; SDKY=0;              TBL01450
         GOWVSAX=0; GOWVSOS=0;                                                 TBL01460
END SUMMARY;                                                                   TBL01470
MAXMINA : PROCEDURE (CA);                                                      TBL01480
DECLARE CA FIXED BINARY (31);                                                  TBL01490
    IF TA      < MINLDAASHO(CA) THEN MINLDAASHO(CA) = TA;                      TBL01500
    IF TA      > MAXLDAASHO(CA) THEN MAXLDAASHO(CA) = TA;                      TBL01510
    MEANAASHO(CA) = MEANAASHO(CA) + TA;                                        TBL01520
    SDAASHO(CA) = SDAASHO(CA) + TA      ** 2;                                  TBL01530
    TA = 0.0;                                                                  TBL01540
END MAXMINA;                                                                   TBL01550
MAXMINK : PROCEDURE(CA);                                                       TBL01560
DECLARE CA FIXED BINARY (31);                                                  TBL01570
    IF TK     < MINLOADKY(CA) THEN MINLOADKY(CA) = TK;                         TBL01580
    IF TK     > MAXLOADKY(CA) THEN MAXLOADKY(CA) = TK;                         TBL01590
    MEANLOADKY(CA) = MEANLOADKY(CA) + TK;                                      TBL01600
    SDKY(CA) = SDKY(CA) + TK     ** 2;                                         TBL01610
    TK = 0.0;                                                                  TBL01620
END MAXMINK;                                                                   TBL01630
GOWCHK:PROCEDURE (W);                                                          TBL01640
DECLARE W FIXED BINARY (31);                                                   TBL01650
    IF VEHWT < 4.0 THEN DO; W=1; GO TO FINISH; END;                            TBL01660
    IF VEHWT>=4.0 &VEHWT<10. THEN DO; W= 2 ;GO TO FINISH; END;                 TBL01670
    IF VEHWT>=10. &VEHWT<15. THEN DO; W=  3;GO TO FINISH; END;                 TBL01680
    IF VEHWT>=15. &VEHWT<20. THEN DO; W=  4;GO TO FINISH; END;                 TBL01690
    IF VEHWT>=20. &VEHWT<22. THEN DO; W=  5;GO TO FINISH; END;                 TBL01700
    IF VEHWT>=22. &VEHWT<24. THEN DO; W=  6;GO TO FINISH; END;                 TBL01710
    IF VEHWT>=24. &VEHWT<26. THEN DO; W=  7;GO TO FINISH; END;                 TBL01720
    IF VEHWT>=26. &VEHWT<28. THEN DO; W=  8;GO TO FINISH; END;                 TBL01730
    IF VEHWT>=28. &VEHWT<30. THEN DO; W=  9;GO TO FINISH; END;                 TBL01740
    IF VEHWT>=30. &VEHWT<32. THEN DO; W= 10;GO TO FINISH; END;                 TBL01750
    IF VEHWT>=32. &VEHWT<34. THEN DO; W= 11;GO TO FINISH; END;                 TBL01760
    IF VEHWT>=34. &VEHWT<36. THEN DO; W= 12;GO TO FINISH; END;                 TBL01770
    IF VEHWT>=36. &VEHWT<38. THEN DO; W= 13;GO TO FINISH; END;                 TBL01780
    IF VEHWT>=38. &VEHWT<40. THEN DO; W= 14;GO TO FINISH; END;                 TBL01790
    IF VEHWT>=40. &VEHWT<45. THEN DO; W= 15;GO TO FINISH; END;                 TBL01800
    IF VEHWT>=45. &VEHWT<50. THEN DO; W= 16;GO TO FINISH; END;                 TBL01810
    IF VEHWT>=50. &VEHWT<55. THEN DO; W= 17;GO TO FINISH; END;                 TBL01820
    IF VEHWT>=55. &VEHWT<60. THEN DO; W= 18;GO TO FINISH; END;                 TBL01830
    IF VEHWT>=60. &VEHWT<65. THEN DO; W= 19;GO TO FINISH; END;                 TBL01840
    IF VEHWT>=65. &VEHWT<70. THEN DO; W= 20;GO TO FINISH; END;                 TBL01850
    IF VEHWT>=70. &VEHWT<75. THEN DO; W= 21;GO TO FINISH; END;                 TBL01860
```

```
    IF VEHWT>=75. &VEHWT<80. THEN DO; W= 22;GO TO FINISH; END;          TBL01870
    IF VEHWT>=80. &VEHWT<85. THEN DO; W= 23;GO TO FINISH; END;          TBL01880
    IF VEHWT>=85. &VEHWT<90. THEN DO; W= 24;GO TO FINISH; END;          TBL01890
    IF VEHWT>=90. &VEHWT<95. THEN DO; W= 25;GO TO FINISH; END;          TBL01900
    W=26;                                                               TBL01910
FINISH: END GOWCHK;                                                     TBL01920
SPDCHK:PROCEDURE(SPD);                                                  TBL01930
DECLARE SPD FIXED BINARY (31);                                          TBL01940
    IF STORE(1,2) < 20.0 THEN DO; SPD=1; GO TO FINISH; END;             TBL01950
    IF STORE(1,2)>=20.&STORE(1,2)<40. THEN DO; SPD=2;GO TO FINISH; END;TBL01960
    IF STORE(1,2)>=40.&STORE(1,2)<50. THEN DO; SPD=3;GO TO FINISH; END;TBL01970
    IF STORE(1,2)>=50.&STORE(1,2)<55. THEN DO; SPD=4;GO TO FINISH; END;TBL01980
    IF STORE(1,2)>=55.&STORE(1,2)<60. THEN DO; SPD=5;GO TO FINISH; END;TBL01990
    IF STORE(1,2)>=60.&STORE(1,2)<65. THEN DO; SPD=6;GO TO FINISH; END;TBL02000
    IF STORE(1,2)>=65.&STORE(1,2)<70. THEN DO; SPD=7;GO TO FINISH; END;TBL02010
    IF STORE(1,2)>=70.&STORE(1,2)<80. THEN DO; SPD=8;GO TO FINISH; END;TBL02020
    IF STORE(1,2)>=80.&STORE(1,2)<90. THEN DO; SPD=9;GO TO FINISH; END;TBL02030
    SPD=10;                                                             TBL02040
FINISH: END SPDCHK;                                                     TBL02050
AXLOAD : PROCEDURE (LOAD,AL);                                           TBL02060
DECLARE LOAD FLOAT BIN(16), AL FIXED BINARY (31);                       TBL02070
    IF LOAD < 1.0 THEN DO;AL=1;LOADKY=0.;LDAASHO=0.;GO TO FINISH;END;   TBL02080
    IF LOAD>=1.0&LOAD<3.0 THEN DO;AL= 2;LOADKY=0.0;LDAASHO=     2.E-4;  TBL02090
       GO TO FINISH; END;                                               TBL02100
    IF LOAD>=3.0&LOAD<5.0 THEN DO;AL=3 ;LOADKY=0.;LDAASHO=      2.E-3;  TBL02110
       GO TO FINISH; END;                                               TBL02120
    IF LOAD>=5.0&LOAD<7.0 THEN DO;AL= 4;LOADKY=0;LDAASHO=      1.E-2;   TBL02130
       GO TO FINISH; END;                                               TBL02140
    IF LOAD>=7.0&LOAD<9.0 THEN DO;AL= 5;LOADKY=0.;LDAASHO=      3.0E-2; TBL02150
       GO TO FINISH; END;                                               TBL02160
    IF LOAD>=9.0&LOAD<11. THEN DO;AL= 6;LOADKY=1.00;LDAASHO=      .09;  TBL02170
       GO TO FINISH; END;                                               TBL02180
    IF LOAD>=11.&LOAD<13. THEN DO;AL= 7;LOADKY=       2.;               TBL02190
       LDAASHO=      .19; GO TO FINISH; END;                           TBL02200
    IF LOAD>=13.&LOAD<15. THEN DO;AL= 8;LOADKY=       4.0;              TBL02210
       LDAASHO=      .36; GO TO FINISH; END;                           TBL02220
    IF LOAD>=15.&LOAD<17. THEN DO;AL= 9;LOADKY=     8.0;                TBL02230
       LDAASHO=      .62; GO TO FINISH; END;                           TBL02240
    IF LOAD>=17.&LOAD<19. THEN DO;AL=10;LOADKY=     16.;                TBL02250
       LDAASHO=1.00; GO TO FINISH; END;                                 TBL02260
    IF LOAD>=19.&LOAD<21. THEN DO;AL=11;LOADKY=     32.;                TBL02270
       LDAASHO=      1.51;  GO TO FINISH; END;                         TBL02280
    IF LOAD>=21.&LOAD<23. THEN DO;AL=12;LOADKY=     64.  ;              TBL02290
       LDAASHO=      2.18;  GO TO FINISH; END;                         TBL02300
    IF LOAD>=23.&LOAD<25. THEN DO;AL=13;LOADKY=     128.;               TBL02310
       LDAASHO=      3.03;  GO TO FINISH; END;                         TBL02320
    IF LOAD>=25.&LOAD<27. THEN DO;AL=14;LOADKY=     256.;               TBL02330
       LDAASHO=      4.09;  GO TO FINISH; END;                         TBL02340
    IF LOAD>=27.&LOAD<29. THEN DO;AL=15;LOADKY=     512.;               TBL02350
       LDAASHO=      5.39;  GO TO FINISH; END;                         TBL02360
    IF LOAD>=29.&LOAD<31. THEN DO;AL=16;GO TO A; END;                   TBL02370
    IF LOAD>=31.&LOAD<33. THEN DO;AL=17;GO TO A; END;                   TBL02380
    IF LOAD>=33.&LOAD<35. THEN DO;AL=18;GO TO A; END;                   TBL02390
    AL=19;  A : LOADKY=      1024.; LDAASHO=     6.97;                  TBL02400
FINISH : END AXLOAD;                                                    TBL02410
ADDAXL : PROCEDURE (CAA,CAD,LD);                                        TBL02420
DECLARE  (CAA,CAD) FIXED BINARY(31), LD FLOAT BINARY(16);               TBL02430
    AXLDVSAX(CAA,CAD) = AXLDVSAX(CAA,CAD) + 1;                          TBL02440
    AXLDVSAX(30 ,CAD) = AXLDVSAX(30 ,CAD) + 1;                          TBL02450
    AXLDVSAX(CAA,20 ) = AXLDVSAX(CAA,20 ) + 1;                          TBL02460
    SDLDAX(1,CAA) = SDLDAX(1,CAA) + LD;                                 TBL02470
```

```
    SDLDAX(2,CAA) = SDLDAX(2,CAA) + LD ** 2;                                TBL02480
END ADDAXL;                                                                 TBL02490
ADAXLAA : PROCEDURE (CAA,CAD,LD);                                           TBL02500
DECLARE  (CAA,CAD) FIXED BINARY(31), LD FLOAT BINARY(16);                   TBL02510
    AXLDVSAXAA(CAA,CAD) = AXLDVSAXAA(CAA,CAD) + 1;                          TBL02520
    AXLDVSAXAA(30 ,CAD) = AXLDVSAXAA(30 ,CAD) + 1;                          TBL02530
    AXLDVSAXAA(CAA,20 ) = AXLDVSAXAA(CAA,20 ) + 1;                          TBL02540
    SDLDAXAA(1,CAA) = SDLDAXAA(1,CAA) + LD;                                 TBL02550
    SDLDAXAA(2,CAA) = SDLDAXAA(2,CAA) + LD ** 2;                            TBL02560
END ADAXLAA;                                                                TBL02570
ADDVEH : PROCEDURE (CA,SPD,W);                                              TBL02580
DECLARE (CA,SPD,W) FIXED BINARY (31);                                       TBL02590
    OSVSAX(30,SPD) = OSVSAX(30,SPD)+1;OSVSAX(CA,11) = OSVSAX(CA,11) +1;TBL02600
    GOWVSAX(30,W) = GOWVSAX(30,W)+1; GOWVSAX(CA,27) = GOWVSAX(CA,27)+1;TBL02610
    OSVSAX(CA,SPD)= OSVSAX(CA,SPD)+1;GOWVSAX(CA,W) =  GOWVSAX(CA,W) +1;TBL02620
    GOWVSOS(SPD,27) = GOWVSOS(SPD,27)+1;GOWVSOS(11,W)=GOWVSOS(11,W) +1;TBL02630
    GOWVSOS(SPD,W) = GOWVSOS(SPD,W) + 1;                                    TBL02640
    SDGOWOS(1,SPD)= SDGOWOS(1,SPD)+ VEHWT;                                  TBL02650
    SDGOWOS(2,SPD)= SDGOWOS(2,SPD)+ VEHWT ** 2;                             TBL02660
    SDOSAX(1,CA) = SDOSAX(1,CA) + STORE(1,2);                               TBL02670
    SDOSAX(2,CA) = SDOSAX(2,CA) + STORE(1,2) ** 2;                          TBL02680
    SDGOWAX(1,CA) = SDGOWAX(1,CA) + VEHWT;                                  TBL02690
    SDGOWAX(2,CA) = SDGOWAX(2,CA) + VEHWT ** 2;                             TBL02700
END ADDVEH;                                                                 TBL02710
TANDAA : PROCEDURE (LOAD);                                                  TBL02720
DECLARE LOAD FLOAT BINARY (16);                                             TBL02730
    IF LOAD < 10.0 THEN DO; LDAASHO = 0.0; GO TO  F; END;                   TBL02740
    IF LOAD >= 10.0 & LOAD < 14.0 THEN DO;LDAASHO=     .01;GO TO F;END;TBL02750
    IF LOAD >= 14.0 & LOAD < 18.0 THEN DO;LDAASHO=     .05;GO TO F;END;TBL02760
    IF LOAD >= 18.0 & LOAD < 22.0 THEN DO;LDAASHO=     .12;GO TO F;END;TBL02770
    IF LOAD >= 22.0 & LOAD < 26.0 THEN DO;LDAASHO=     .26;GO TO F;END;TBL02780
    IF LOAD >= 26.0 & LOAD < 30.0 THEN DO;LDAASHO=     .50;GO TO F;END;TBL02790
    IF LOAD >= 30.0 & LOAD < 34.0 THEN DO;LDAASHO=     .81;GO TO F;END;TBL02800
    IF LOAD >= 34.0 & LOAD < 38. THEN DO;LDAASHO=     1.38;GO TO F;END;TBL02810
    IF LOAD >= 38.0 & LOAD < 42. THEN DO;LDAASHO=     2.08;GO TO F;END;TBL02820
    IF LOAD >= 42.0 & LOAD < 46. THEN DO;LDAASHO=     3.00;GO TO F;END;TBL02830
    IF LOAD >= 46.0 & LOAD < 50. THEN DO;LDAASHO=     4.17;GO TO F;END;TBL02840
    IF LOAD >= 50.0 & LOAD < 54. THEN DO;LDAASHO=     5.63;GO TO F;END;TBL02850
    IF LOAD >= 54.0 & LOAD < 58. THEN DO;LDAASHO=     7.41;GO TO F;END;TBL02860
    LDAASHO =        9.59;                                                  TBL02870
F  :END TANDAA;                                                             TBL02880
ADDVHKY : PROCEDURE (CA);                                                   TBL02890
DECLARE CA FIXED BINARY (31);                                               TBL02900
    AXLDVSAX(CA,21) = AXLDVSAX(CA,21) + 1;                                  TBL02910
END ADDVHKY;                                                                TBL02920
ADDVHAA : PROCEDURE(CA);                                                    TBL02930
DECLARE CA FIXED BINARY (31);                                               TBL02940
    AXLDVSAXAA(CA,21) = AXLDVSAXAA(CA,21) + 1;                              TBL02950
END ADDVHAA;                                                                TBL02960
AXLE2L : PROCEDURE;                                                         TBL02970
    CALL ADDVEH(1,SPD,W); CALL ADDVHKY(1); CALL ADDVHAA(1);                 TBL02980
    DO I=1 TO 2;  CALL AXLOAD(STORE(I,1),AL);                               TBL02990
         CALL ADDAXL(1,AL,STORE(I,1)); TA=TA+LDAASHO; TK=TK+LOADKY;         TBL03000
         CALL ADAXLAA(1,AL,STORE(I,1));  END;                               TBL03010
         CALL MAXMINA(1);  CALL MAXMINK(1);                                 TBL03020
END AXLE2L;                                                                 TBL03030
AXLE2H : PROCEDURE;                                                         TBL03040
    CALL ADDVEH(2,SPD,W); CALL ADDVHKY(2); CALL ADDVHAA(2);                 TBL03050
    DO I=1 TO 2;  CALL AXLOAD(STORE(I,1),AL);                               TBL03060
         CALL ADDAXL(2,AL,STORE(I,1)); TA=TA+LDAASHO;  TK=TK+LOADKY;        TBL03070
         CALL ADAXLAA(2,AL,STORE(I,1));  END;                               TBL03080
```

```
        CALL MAXMINA(2);  CALL MAXMINK(2);                                  TBL03090
END AXLE2H;                                                                 TBL03100
AXLE3 : PROCEDURE;                                                          TBL03110
    IF STORE(3,3) <= 3.33 THEN DO; CALL ADDVHAA(4); CALL ADDVEH(4,SPD,      TBL03120
        W); CALL AXLOAD(STORE(1,1),AL); TA=TA+LDAASHO;                      TBL03130
        DWT = STORE(2,1) + STORE(3,1); CALL ADAXLAA(4,AL,STORE(1,1));       TBL03140
        CALL AXLOAD(DWT,AL); CALL ADAXLAA(4,AL,DWT); CALL TANDAA(DWT);      TBL03150
        TA=TA+LDAASHO;       CALL MAXMINA(4); RETURN; END;                  TBL03160
    CALL ADDVEH(3,SPD,W); CALL ADDVHAA(3); DO I=1 TO 3; CALL AXLOAD         TBL03170
        (STORE(I,1),AL); CALL ADAXLAA(3,AL,STORE(I,1));TA=TA+LDAASHO;       TBL03180
        END; CALL MAXMINA(3);                                               TBL03190
    IF STORE(3,3) > 10.0 THEN DO; CALL ADDVHKY(3); DO I=1 TO 3; CALL        TBL03200
        AXLOAD(STORE(I,1),AL); CALL ADDAXL(3,AL,STORE(I,1));                TBL03210
        TK=TK+LOADKY;  END;  CALL MAXMINK(3); RETURN; END;                  TBL03220
    CALL ADDVHKY(4); CALL AXLOAD(STORE(1,1),AL); TK=TK+LOADKY;              TBL03230
        CALL ADDAXL(4,AL,STORE                                              TBL03240
        (1,1)); DWT = STORE(2,1) + STORE(3,1); CALL AXLOAD(DWT,AL);         TBL03250
        CALL ADDAXL(4,AL,DWT); TK=TK+LOADKY; CALL MAXMINK(4);               TBL03260
END AXLE3;                                                                  TBL03270
AXLE4 : PROCEDURE;                                                          TBL03280
    IF STORE(3,3) <= 3.33 & STORE(4,3) <= 3.33 THEN DO; CALL ADDVHAA        TBL03290
        (8); CALL ADDVEH(8,SPD,W); CALL AXLOAD(STORE(1,1),AL); CALL         TBL03300
        ADAXLAA(8,AL,STORE(1,1)); TA=TA+LDAASHO;   DWT = STORE(2,1) +       TBL03310
        STORE(3,1) + STORE(4,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA          TBL03320
        (8,AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO; CALL MAXMINA(8);       TBL03330
        RETURN;  END;                                                       TBL03340
    IF STORE(3,3) <= 3.33 THEN DO; CALL ADDVHAA(6); CALL ADDVEH(6,SPD,      TBL03350
        W); CALL AXLOAD(STORE(1,1),AL); TA=TA+LDAASHO;    CALL            TBL03360
        ADAXLAA(6,AL,STORE(1,1)); CALL AXLOAD(STORE(4,1),AL);               TBL03370
        CALL ADAXLAA(6,AL,STORE(4,1)); TA=TA+LDAASHO;   DWT =               TBL03380
        STORE(2,1) + STORE(3,1); CALL AXLOAD(DWT,AL); CALL TANDAA           TBL03390
        (DWT); CALL ADAXLAA(6,AL,DWT); TA=TA+LDAASHO; CALL MAXMINA(6);      TBL03400
        RETURN; END;                                                        TBL03410
    IF STORE(4,3) <= 3.33 THEN DO; CALL ADDVHAA(7); CALL ADDVEH(7,SPD,      TBL03420
        W); CALL AXLOAD(STORE(1,1),AL); TA=TA+LDAASHO;   CALL ADAXLAA       TBL03430
        (7,AL,STORE(1,1)); CALL AXLOAD(STORE(2,1),AL); CALL ADAXLAA         TBL03440
        (7,AL,STORE(2,1)); TA=TA+LDAASHO;   DWT = STORE(3,1) + STORE        TBL03450
        (4,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(7,AL,DWT); CALL            TBL03460
        TANDAA(DWT); TA=TA+LDAASHO; CALL MAXMINA(7);  RETURN; END;          TBL03470
    CALL ADDVHAA(5); CALL ADDVEH(5,SPD,W); DO I=1 TO 4; CALL AXLOAD         TBL03480
        (STORE(I,1),AL); CALL  ADAXLAA(5,AL,STORE(I,1));TA=TA+LDAASHO;      TBL03490
        END; CALL MAXMINA(5);                                               TBL03500
    IF STORE(3,3) <= 10.0 & STORE(4,3) <= 10.0 THEN DO; CALL ADDVHKY        TBL03510
        (8); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(8,AL,STORE(1,1));      TBL03520
        TK=TK+LOADKY;                                                       TBL03530
        DWT = STORE(2,1) + STORE (3,1) + STORE(4,1); CALL AXLOAD            TBL03540
        (DWT,AL); CALL ADDAXL(8,AL,DWT); TK=TK+LOADKY;CALL MAXMINK(8);      TBL03550
        RETURN; END;                                                        TBL03560
    IF STORE(3,3) <= 10.0 THEN DO; CALL ADDVHKY(6); CALL AXLOAD(STORE       TBL03570
        (1,1),AL); CALL ADDAXL(6,AL,STORE(1,1)); TK=TK+LOADKY;              TBL03580
                                                 CALL AXLOAD(STORE          TBL03590
        (4,1),AL); CALL ADDAXL(6,AL,STORE(4,1)); TK=TK+LOADKY;              TBL03600
                                                 DWT = STORE(2,1) +         TBL03610
        STORE(3,1); CALL AXLOAD(DWT,AL); TK=TK+LOADKY; CALL ADDAXL          TBL03620
        (6,AL,DWT); CALL MAXMINK(6); RETURN; END;                           TBL03630
    IF STORE(4,3) <= 10.0 THEN DO; CALL ADDVHKY(7); CALL AXLOAD(STORE       TBL03640
        (1,1),AL); CALL ADDAXL(7,AL,STORE(1,1)); TK=TK+LOADKY;              TBL03650
                                                 CALL AXLOAD(STORE          TBL03660
        (2,1),AL); CALL ADDAXL(7,AL,STORE(2,1)); TK=TK+LOADKY;              TBL03670
                                                 DWT = STORE(3,1)  +        TBL03680
        STORE(4,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(7,AL,DWT);             TBL03690
```

```
      TK=TK+LOADKY; CALL MAXMINK(7); RETURN; END;                          TBL03700
    CALL ADDVHKY(5); DO I=1 TO 4; CALL AXLOAD(STORE(I,1),AL); CALL         TBL03710
      ADDAXL(5,AL,STORE(I,1)); TK=TK+LOADKY; END; CALL MAXMINK(5);         TBL03720
END AXLE4;                                                                 TBL03730
AXLE5 : PROCEDURE;                                                         TBL03740
    IF STORE(4,3)<=3.33 & STORE(5,3)<=3.33 THEN DO;                        TBL03750
      CALL ADDVHAA(14);CALL ADDVEH(14,SPD,W); DO I=1 TO 2; CALL            TBL03760
      AXLOAD(STORE(I,1),AL);CALL ADAXLAA(14,AL,STORE(I,1)); TA=            TBL03770
      TA+LDAASHO;  END; DWT=STORE(3,1)+STORE(4,1)+STORE(5,1);              TBL03780
      CALL AXLOAD(DWT,AL); CALL ADAXLAA(14,AL,DWT);CALL TANDAA(DWT);TBL03790
      TA=TA+LDAASHO; CALL MAXMINA(14); RETURN; END;                        TBL03800
    IF STORE(3,3)<=3.33 & STORE(4,3)<=3.33 THEN DO;                        TBL03810
      CALL ADDVHAA(13); CALL ADDVEH(13,SPD,W);CALL AXLOAD(STORE(1,1)TBL03820
      ,AL); CALL ADAXLAA(13,AL,STORE(1,1)); TA=TA+LDAASHO;                 TBL03830
      DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT,AL);CALLTBL03840
      ADAXLAA(13,AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;    CALL        TBL03850
      AXLOAD(STORE(5,1),AL);CALL ADAXLAA(13,AL,STORE(5,1));                TBL03860
      TA=TA+LDAASHO; CALL MAXMINA(13); RETURN; END;                        TBL03870
    IF STORE(3,3)<=3.33 & STORE(5,3)<=3.33 THEN DO;                        TBL03880
      CALL ADDVHAA(15); CALL ADDVEH(15,SPD,W);CALL AXLOAD(STORE(1,1)TBL03890
      ,AL); CALL ADAXLAA(15,AL,STORE(1,1)); TA=TA+LDAASHO;                 TBL03900
      DO I=2 TO 4 BY 2; DWT=STORE(I,1)+STORE(I+1,1); CALL AXLOAD           TBL03910
      (DWT,AL); CALL ADAXLAA(15,AL,DWT);CALL TANDAA(DWT);                  TBL03920
      TA=TA+LDAASHO; END; CALL MAXMINA(15); RETURN; END;                   TBL03930
    IF STORE(5,3)<=3.33 THEN DO;                                           TBL03940
      CALL ADDVHAA(12);CALL ADDVEH(12,SPD,W);DO I=1 TO 3; CALL             TBL03950
      AXLOAD(STORE(I,1),AL);CALL ADAXLAA(12,AL,STORE(I,1)); TA=            TBL03960
      TA+LDAASHO;  END; DWT=STORE(4,1)+STORE(5,1); CALL AXLOAD(DWT,TBL03970
      AL); CALL ADAXLAA(12,AL,DWT); CALL TANDAA(DWT);                      TBL03980
      TA=TA+LDAASHO; CALL MAXMINA(12); RETURN; END;                        TBL03990
    IF STORE(4,3)<=3.33 THEN DO;                                           TBL04000
      CALL ADDVHAA(11); CALL ADDVEH(11,SPD,W); DO I=1 TO 2; CALL           TBL04010
      AXLOAD(STORE(I,1),AL);CALL ADAXLAA(11,AL,STORE(I,1)); TA=            TBL04020
      TA+LDAASHO;  END; DWT=STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT,TBL04030
      AL);CALL ADAXLAA(11,AL,DWT); CALL TANDAA(DWT);TA=TA+LDAASHO;TBL04040
           CALL AXLOAD(STORE(5,1),AL); CALL ADAXLAA(11,AL,STORE(5,1TBL04050
      ));  TA=TA+LDAASHO; CALL MAXMINA(11); RETURN; END;                    TBL04060
    IF STORE(3,3)<=3.33 THEN DO;                                           TBL04070
      CALL ADDVHAA(10);CALL AODVEH(10,SPD,W); CALL AXLOAD(STORE(1,1)TBL04080
      ,AL); CALL ADAXLAA(10,AL,STORE(1,1)); TA=TA+LDAASHO;    DWT=  TBL04090
      STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(10,TBL04100
      AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;    DO I=4 TO 5; CALLTBL04110
      AXLDAD(STORE(I,1),AL); CALL ADAXLAA(10,AL,STORE(I,1));               TBL04120
      TA=TA+LDAASHO; END; CALL MAXMINA(10); RETURN; END;                   TBL04130
    CALL ADDVHAA(9); CALL ADDVEH(9,SPD,W); DO I=1 TO 5; CALL AXLOAD        TBL04140
      (STORE(I,1),AL); CALL ADAXLAA(9,AL,STORE(I,1));TA=TA+LDAASHO; TBL04150
      END; CALL MAXMINA(9);                                                TBL04160
    IF STORE(4,3)<=10.0 & STORE(5,3)<=10.0 THEN DO;                        TBL04170
      CALL ADDVHKY(14); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);           TBL04180
      CALL ADDAXL(14,AL,STORE(I,1)); TK=TK+LOADKY;                         TBL04190
                                     END; DWT=STORE(3,1)+STORE(4,1) TBL04200
      +STORE(5,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(14,AL,DWT);            TBL04210
      TK=TK+LOADKY; CALL MAXMINK(14); RETURN; END;                         TBL04220
    IF STORE(3,3)<=10.0 & STORE(4,3)<=10.0 THEN DO;                        TBL04230
      CALL ADDVHKY(13); CALL AXLOAD(STORE(1,1),AL);CALL ADDAXL(13,         TBL04240
      AL,STORE(1,1)); TK=TK+LOADKY;                                        T8L04250
                      DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL           TBL04260
      AXLOAD(DWT,AL); CALL ADDAXL(13,AL,OWT); TK=TK+LOADKY;                TBL04270
                                              CALL AXLOAD(STORE(5,1)TBL04280
      ,AL); CALL ADDAXL(13,AL,STORE(5,1)); TK=TK+LOADKY;                   TBL04290
      CALL MAXMINK(13); RETURN; END;                                       TBL04300
```

```
    IF STORE(3,3)<=10.0 & STORE(5,3)<=10.0 THEN DO;                          TBL04310
        CALL ADDVHKY(15); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(15, TBL04320
        AL,STORE(1,1)); TK=TK+LOADKY;                                        TBL04330
                        DO I=2 TO 4 BY 2; DWT=STORE(I,1)+STORE(I+1,1);TBL04340
        CALL AXLOAD(DWT,AL); CALL ADDAXL(15,AL,DWT); TK=TK+LOADKY;           TBL04350
        END; CALL MAXMINK(15); RETURN; END;                                  TBL04360
    IF STORE(5,3)<=10.0 THEN DO;                                             TBL04370
        CALL ADDVHKY(12); DO I=1 TO 3; CALL AXLOAD(STORE(I,1),AL);CALLTBL04380
        ADDAXL(12,AL,STORE(I,1)); TK=TK+LOADKY;                              TBL04390
                                  END; DWT=STORE(4,1)+STORE(5,1); CALLTBL04400
        AXLOAD(DWT,AL);CALL ADDAXL(12,AL,DWT); TK=TK+LOADKY; CALL      TBL04410
        MAXMINK(12); RETURN; END;                                            TBL04420
    IF STORE(4,3)<=10.0 THEN DO;                                             TBL04430
        CALL ADDVHKY(11); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);CALLTBL04440
        ADDAXL(11,AL,STORE(I,1)); TK=TK+LOADKY;                              TBL04450
                                  END; DWT=STORE(3,1)+STORE(4,1); CALLTBL04460
        AXLOAD(DWT,AL); TK=TK+LOADKY;                                        TBL04470
                        CALL ADDAXL(11,AL,DWT); CALL AXLOAD(STORE(5,1)TBL04480
        ,AL);CALL ADDAXL(11,AL,STORE(5,1)); TK=TK+LOADKY;                    TBL04490
        CALL MAXMINK(11); RETURN; END;                                       TBL04500
    IF STORE(3,3)<=10.0 THEN DO;                                             TBL04510
        CALL ADDVHKY(10); CALL AXLOAD(STORE(1,1),AL);CALL ADDAXL(10,  TBL04520
        AL,STORE(1,1)); TK=TK+LOADKY;                                        TBL04530
                        DWT=STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT,   TBL04540
        AL); TK=TK+LOADKY;                                                   TBL04550
             CALL ADDAXL(10,AL,DWT); DO I=4 TO 5; CALL AXLOAD(STORE   TBL04560
        (I,1),AL);CALL ADDAXL(10,AL,STORE(I,1)); TK=TK+LOADKY; END;   TBL04570
        CALL MAXMINK(10); RETURN; END;                                       TBL04580
    CALL ADDVHKY(9); DO I=1 TO 5; CALL AXLOAD(STORE(I,1),AL); CALL       TBL04590
        ADDAXL(9,AL,STORE(I,1)); TK=TK+LOADKY; END; CALL MAXMINK(9);  TBL04600
    RETURN;                                                                  TBL04610
END AXLE5;                                                                   TBL04620
AXLE6 : PROCEDURE;                                                           TBL04630
    IF STORE(3,3)<=3.33 & STORE(5,3)<=3.33 & STORE(6,3)<=3.33 THEN DO; TBL04640
        CALL ADDVHAA(25); CALL ADDVEH(25,SPD,W);CALL AXLOAD(STORE(1,1)TBL04650
        ,AL); CALL ADAXLAA(25,AL,STORE(1,1)); TA=TA+LDAASHO;    DWT=  TBL04660
        STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(25,  TBL04670
        AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;    DWT=STORE(4,1)+  TBL04680
        STORE(5,1)+STORE(6,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(25,  TBL04690
        AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO; CALL MAXMINA(25);   TBL04700
        RETURN; END;                                                         TBL04710
    IF STORE(3,3)<=3.33 & STORE(4,3)<=3.33 & STORE(6,3)<=3.33 THEN DO; TBL04720
        CALL ADDVHAA(24); CALL ADDVEH(24,SPD,W); CALL AXLOAD(STORE(1, TBL04730
        1),AL); CALL ADAXLAA(24,AL,STORE(1,1));TTA=TA+LDAASHO;    DWT=TBL04740
        STORE(2,1)+STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT,AL); CALL   TBL04750
        ADAXLAA(24,AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;    DWT=  TBL04760
        STORE(5,1)+STORE(6,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(24,  TBL04770
        AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO; CALL MAXMINA(24);   TBL04780
        RETURN; END;                                                         TBL04790
    IF STORE(3,3)<=3.33 & STORE(6,3)<=3.33 THEN DO;                          TBL04800
        CALL ADDVHAA(23); CALL ADDVEH(23,SPD,W); DO I=1 TO 4 BY 3;CALLTBL04810
        AXLOAD(STORE(I,1),AL); CALL ADAXLAA(23,AL,STORE(I,1)); TA=    TBL04820
        TA+LDAASHO;  DWT=STORE(I+1,1)+STORE(I+2,1); CALL AXLOAD(DWT,  TBL04830
        AL); CALL ADAXLAA(23,AL,DWT); CALL TANDAA(DWT);                      TBL04840
        TA=TA+LDAASHO; END; CALL MAXMINA(23); RETURN; END;                   TBL04850
    IF STORE(4,3)<=3.33 & STORE(6,3)<=3.33 THEN DO;                          TBL04860
        CALL ADDVHAA(22); CALL ADDVEH(22,SPD,W); DO I=1 TO 2; CALL    TBL04870
        AXLOAD(STORE(I,1),AL); CALL ADAXLAA(22,AL,STORE(I,1)); TA=    TBL04880
        TA+LDAASHO;  END; DO I=3 TO 5 BY 2; DWT=STORE(I,1)+STORE(I+1, TBL04890
        1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(22,AL,DWT); CALL TANDAA TBL04900
        (DWT); TA=TA+LDAASHO; END; CALL MAXMINA(22); RETURN; END;        TBL04910
```

```
IF STORE(3,3)<=3.33 & STORE(5,3)<=3.33 THEN DO;                          TBL04920
     CALL ADDVHAA(21); CALL ADDVEH(21,SPD,W); CALL AXLOAD(STORE(1,       TBL04930
     1),AL); CALL ADAXLAA(21,AL,STORE(1,1)); TA=TA+LDAASHO;    DO        TBL04940
     I=2 TO 4 BY 2; DWT=STORE(I,1)+STORE(I+1,1); CALL AXLOAD(DWT,        TBL04950
     AL); CALL ADAXLAA(21,AL,DWT); CALL TANDAA(DWT);TA=TA+LDAASHO;       TBL04960
     END; CALL AXLOAD(STORE(6,1),AL); CALL ADAXLAA(21,AL,                TBL04970
     STORE(6,1)); TA=TA+LDAASHO; CALL MAXMINA(21); RETURN; END;          TBL049B0
IF STORE(6,3)<=3.33 & STORE(5,3)<=3.33 THEN DO;                          TBL04990
     CALL ADDVHAA(28); CALL ADDVEH(28,SPD,W); DO I=1 TO 3; CALL          TBL05000
     AXLOAD(STORE(I,1),AL); CALL ADAXLAA(28,AL,STORE(I,1)); TA=          TBL05010
     TA+LDAASHO;  END; DWT=STORE(4,1)+STORE(5,1)+STORE(6,1); CALL        TBL05020
     AXLOAD(DWT,AL); CALL ADAXLAA(28,AL,DWT); CALL TANDAA(DWT);          TBL05030
     TA=TA+LDAASHO; CALL MAXMINA(28); RETURN; END;                       TBL05040
IF STORE(5,3)<=3.33 & STORE(4,3)<=3.33 THEN DO;                          TBL05050
     CALL ADDVHAA(27); CALL ADDVEH(27,SPD,W); DO I=1 TO 2; CALL          TBL05060
     AXLOAD(STORE(I,1),AL); CALL ADAXLAA(27,AL,STORE(I,1)); TA=          TBL05070
     TA+LDAASHO;  END; DWT=STORE(3,1)+STORE(4,1)+STORE(5,1); CALL        TBL05080
     AXLOAD(DWT,AL); CALL ADAXLAA(27,AL,DWT); CALL TANDAA(DWT);          TBL05090
     TA=TA+LDAASHO;    CALL AXLOAD(STORE(6,1),AL); CALL ADAXLAA(27,      TBL05100
     AL,STORE(6,1)); TA=TA+LDAASHO; CALL MAXMINA(27); RETURN; END;       TBL05110
IF STORE(4,3)<=3.33 & STORE(3,3)<=3.33 THEN DO;                          TBL05120
     CALL ADDVHAA(26); CALL ADDVEH(26,SPD,W); CALL AXLOAD(STORE(1,       TBL05130
     1),AL); CALL ADAXLAA(26,AL,STORE(1,1)); TA=TA+LDAASHO;    DWT=      TBL05140
     STORE(2,1)+STORE(3,1)+STORE (4,1); CALL AXLOAD(DWT,AL); CALL        TBL05150
     ADAXLAA(26,AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;    DO I=5      TBL05160
     TO 6; CALL AXLOAD(STORE(I,1),AL); CALL ADAXLAA(26,AL,STORE(I,       TBL05170
     1)); TA=TA+LDAASHO; END; CALL MAXMINA(26); RETURN; END;             TBL05180
IF STORE(6,3)<=3.33 THEN DO;                                             TBL05190
     CALL ADDVHAA(20); CALL ADDVEH(20,SPD,W); DO I=1 TO 4; CALL          TBL05200
     AXLOAD(STORE(I,1),AL); CALL ADAXLAA(20,AL,STORE(I,1)); TA=          TBL05210
     TA+LDAASHO;  END; DWT=STORE(5,1)+STORE(6,1); CALL AXLOAD(DWT,       TBL05220
     AL); CALL ADAXLAA(20,AL,DWT); CALL TANDAA(DWT);                     TBL05230
     TA=TA+LDAASHO; CALL MAXMINA(20); RETURN; END;                       TBL05240
IF STORE(5,3)<=3.33 THEN DO;                                             TBL05250
     CALL ADDVHAA(19); CALL ADDVEH(19,SPD,W); DO I=1 TO 3; CALL          TBL05260
     AXLOAD(STORE(I,1),AL); CALL ADAXLAA(19,AL,STORE(I,1)); TA=          TBL05270
     TA+LDAASHO;  END; DWT=STORE(4,1)+STORE(5,1); CALL AXLOAD(DWT,       TBL052B0
     AL); CALL ADAXLAA(19,AL,DWT); CALL TANDAA(DWT);TA=TA+LDAASHO;       TBL05290
          CALL AXLOAD(STORE(6,1),AL);  CALL ADAXLAA(19,AL,STORE(6,       TBL05300
     1)); CALL MAXMINA(19); RETURN; END;                                 TBL05310
IF STORE(4,3)<=3.33 THEN DO;                                             TBL05320
     CALL ADDVHAA(18); CALL ADDVEH(18,SPD,W); DO I=1 TO 2; CALL          TBL05330
     AXLOAD(STORE(I,1),AL); CALL ADAXLAA(18,AL,STORE(I,1)); TA=          TBL05340
     TA+LDAASHO;  END; DWT=STORE(3,1)+STORE(4,1); CALL AXLOAD(DWT,       TBL05350
     AL); CALL ADAXLAA(18,AL,DWT); CALL TANDAA(DWT);TA=TA+LDAASHO;       TBL05360
          DO I=5 TO 6; CALL AXLOAD(STORE(I,1),AL); CALL ADAXLAA          TBL05370
     (18,AL,STORE(I,1)); TA=TA+LDAASHO; END; CALL MAXMINA(18);           TBL053B0
     RETURN; END;                                                        TBL05390
IF STORE(3,3)<=3.33 THEN DO;                                             TBL05400
     CALL ADDVHAA(17); CALL ADDVEH(17,SPD,W); CALL AXLOAD(STORE(1,       TBL05410
     1),AL); CALL ADAXLAA(17,AL,STORE(1,1)); TA=TA+LDAASHO;    DWT=      TBL05420
     STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT,AL); CALL ADAXLAA(17,        TBL05430
     AL,DWT); CALL TANDAA(DWT); TA=TA+LDAASHO;    DO I=4 TO 6; CALL      TBL05440
     AXLOAD(STORE(I,1),AL); CALL ADAXLAA(17,AL,STORE(I,1));              TBL05450
     TA=TA+LDAASHO; END; CALL MAXMINA(17); RETURN; END;                  TBL05460
CALL ADDVHAA(16); CALL ADDVEH(16,SPD,W); DO I=1 TO 6; CALL AXLOAD        TBL05470
     (STORE(I,1),AL); CALL ADAXLAA(16,AL,STORE(I,1));TA=TA+LDAASHO;      TBL054B0
     END; CALL MAXMINA(16);                                              TBL05490
IF STORE(3,3)<=10.0 & STORE(5,3)<=10.0 & STORE(6,3)<=10.0 THEN DO;       TBL05500
     CALL ADDVHKY(25); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(25,       TBL05510
     AL,STORE(1,1)); TK=TK+LOADKY;                                       TBL05520
```

```
                    DWT=STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT,    TBL05530
     AL); TK=TK+LOADKY;                                            TBL05540
          CALL ADDAXL(25,AL,DWT); DWT=STORE(4,1)+STORE(5,1)+STORE  TBL05550
     (6,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(25,AL,DWT);           TBL05560
     TK=TK+LOADKY; CALL MAXMINK(25); RETURN; END;                  TBL05570
IF STORE(3,3)<=10.0 & STORE(4,3)<=10.0 & STORE(6,3)<=10.0 THEN DO; TBL05580
     CALL ADDVHKY(24); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(24, TBL05590
     AL,STORE(1,1));TK=TK+LOADKY;                                  TBL05600
                    DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL     TBL05610
     AXLOAD(DWT,AL); TK=TK+LOADKY;                                 TBL05620
                    CALL ADDAXL(24,AL,DWT); DWT=STORE(5,1)+STORE   TBL05630
     (6,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(24,AL,DWT);           TBL05640
     TK=TK+LOADKY; CALL MAXMINK(24); RETURN; END;                  TBL05650
IF STORE(3,3)<=10.0 & STORE(6,3)<=10.0 THEN DO;                    TBL05660
     CALL ADDVHKY(23); DO I=1 TO 4 BY 3; CALL AXLOAD(STORE(I,1),   TBL05670
     AL); TK=TK+LOADKY;                                            TBL05680
          CALL ADDAXL(23,AL,STORE(I,1)); DWT=STORE(I+1,1)+STORE    TBL05690
     (I+2,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(23,AL,DWT);         TBL05700
     TK=TK+LOADKY; END; CALL MAXMINK(23); RETURN; END;             TBL05710
IF STORE(4,3)<=10.0 & STORE(6,3)<=10.0 THEN DO;                    TBL05720
     CALL ADDVHKY(22); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);CALLTBL05730
     ADDAXL(22,AL,STORE(I,1)); TK=TK+LOADKY;                       TBL05740
                              END; DO I=3 TO 5 BY 2; DWT=STORE(I,  TBL05750
     1)+STORE(I+1,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(22,AL,DWT); TBL05760
     TK=TK+LOADKY; END; CALL MAXMINK(22); RETURN; END;             TBL05770
IF STORE(3,3)<=10.0 & STORE(5,3)<=10.0 THEN DO;                    TBL05780
     CALL ADDVHKY(21); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(21, TBL05790
     AL,STORE(1,1)); TK=TK+LOADKY;                                 TBL05800
                    DO I=2 TO 4 BY 2; DWT=STORE(I,1)+STORE(I+1,1);TBL05810
     CALL AXLOAD(DWT,AL); TK=TK+LOADKY;                            TBL05820
                         CALL ADDAXL(21,AL,DWT); END; CALL AXLOAD  TBL05830
     (STORE(6,1),AL); CALL ADDAXL(21,AL,STORE(6,1));               TBL05840
     TK=TK+LOADKY; CALL MAXMINK(21); RETURN; END;                  TBL05850
IF STORE(6,3)<=10.0 & STORE(5,3)<=10.0 THEN DO;                    TBL05860
     CALL ADDVHKY(28); DO I=1 TO 3; CALL AXLOAD(STORE(I,1),AL);CALLTBL05870
     ADDAXL(28,AL,STORE(I,1)); TK=TK+LOADKY;                       TBL05880
                              END; DWT=STORE(4,1)+STORE(5,1)+      TBL05890
     STORE(6,1); CALL AXLOAD(DWT,AL); CALL ADDAXL(28,AL,DWT);      TBL05900
     TK=TK+LOADKY; CALL MAXMINK(28); RETURN; END;                  TBL05910
IF STORE(5,3)<=10.0 & STORE(4,3)<=10.0 THEN DO;                    TBL05920
     CALL ADDVHKY(27); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);CALLTBL05930
     ADDAXL(27,AL,STORE(I,1)); TK=TK+LOADKY;                       TBL05940
                              END; DWT=STORE(3,1)+STORE(4,1)+STORETBL05950
     (5,1); CALL AXLOAD(DWT,AL); TK=TK+LOADKY;                     TBL05960
                               CALL ADDAXL(27,AL,DWT); CALL        TBL05970
     AXLOAD(STORE(6,1),AL); CALL ADDAXL(27,AL,STORE(6,1));         TBL05980
     TK=TK+LOADKY;      CALL MAXMINK(27); RETURN; END;             TBL05990
IF STORE(4,3)<=10.0 & STORE(3,3)<=10.0 THEN DO;                    TBL06000
     CALL ADDVHKY(26); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(26, TBL06010
     AL,STORE(1,1));TK=TK+LOADKY;                                  TBL06020
                    DWT=STORE(2,1)+STORE(3,1)+STORE(4,1); CALL     TBL06030
     AXLOAD(DWT,AL); TK=TK+LOADKY;                                 TBL06040
                    CALL ADDAXL(26,AL,DWT); DO I=5 TO 6; CALL      TBL06050
     AXLOAD(STORE(I,1),AL); CALL ADDAXL(26,AL,STORE(I,1));         TBL06060
     TK=TK+LOADKY; END; CALL MAXMINK(26); RETURN; END;             TBL06070
IF STORE(6,3)<=10.0 THEN DO;                                       TBL06080
     CALL ADDVHKY(20); DO I=1 TO 4; CALL AXLOAD(STORE(I,1),AL);CALLTBL06090
     ADDAXL(20,AL,STORE(I,1)); TK=TK+LOADKY;                       TBL06100
                              END; DWT=STORE(5,1)+STORE(6,1); CALLTBL06110
     AXLOAD(DWT,AL); CALL ADDAXL(20,AL,DWT);                       TBL06120
     TK=TK+LOADKY; CALL MAXMINK(20); RETURN; END;                  TBL06130
```

```
    IF STORE(5,3)<=10.0 THEN DO;                                          TBL06140
         CALL ADDVHKY(19); DO I=1 TO 3; CALL AXLOAD(STORE(I,1),AL);CALLTBL06150
         ADDAXL(19,AL,STORE(I,1)); TK=TK+LOADKY;                          TBL06160
                                   END; DWT=STORE(4,1)+STORE(5,1); CALLTBL06170
         AXLOAD(DWT,AL); TK=TK+LOADKY;                                    TBL06180
                         CALL ADDAXL(19,AL,DWT); CALL AXLOAD(STORE(6,1)TBL06190
         ,AL); CALL ADDAXL(19,AL,STORE(6,1));                             TBL06200
         TK=TK+LOADKY; CALL MAXMINK(19); RETURN; END;                     TBL06210
    IF STORE(4,3)<=10.0 THEN DO;                                          TBL06220
         CALL ADDVHKY(18); DO I=1 TO 2; CALL AXLOAD(STORE(I,1),AL);CALLTBL06230
         ADDAXL(18,AL,STORE(I,1)); TK=TK+LOADKY;                          TBL06240
                          END;     DWT=STORE(3,1)+STORE(4,1); CALL      TBL06250
         AXLOAD(DWT,AL);TK=TK+LOADKY;                                     TBL06260
                         CALL ADDAXL(18,AL,DWT); DO I=5 TO 6; CALL        TBL06270
         AXLOAD(STORE(I,1),AL); CALL ADDAXL(18,AL,STORE(I,1));            TBL06280
         TK=TK+LOADKY; END; CALL MAXMINK(18); RETURN; END;                TBL06290
    IF STORE(3,3)<=10.0 THEN DO;                                          TBL06300
         CALL ADDVHKY(17); CALL AXLOAD(STORE(1,1),AL); CALL ADDAXL(17,    TBL06310
         AL,STORE(1,1)); TK=TK+LOADKY;                                    TBL06320
                         DWT=STORE(2,1)+STORE(3,1); CALL AXLOAD(DWT,      TBL06330
         AL); TK=TK+LOADKY;                                               TBL06340
              CALL ADDAXL(17,AL,DWT); DO I=4 TO 6; CALL AXLOAD(STORE      TBL06350
         (1,1),AL); CALL ADDAXL(17,AL,STORE(1,1));                        TBL06360
         TK=TK+LOADKY; END; CALL MAXMINK(17); RETURN; END;                TBL06370
    CALL ADDVHKY(16); DO I=1 TO 6; CALL AXLOAD(STORE(I,1),AL); CALL       TBL06380
         ADDAXL(16,AL,STORE(I,1));                                        TBL06390
         TK=TK+LOADKY; END; CALL MAXMINK(16);                             TBL06400
         RETURN;                                                          TBL06410
END AXLE6;                                                                TBL06420
AXLEM6 : PROCEDURE;                                                       TBL06430
    CALL ADDVHAA(29); CALL ADDVHKY(29); CALL ADDVEH(29,SPD,W); DO I=1     TBL06440
         TO NUMBER; CALL AXLOAD(STORE(I,1),AL); CALL ADAXLAA(29,AL,       TBL06450
         STORE(I,1)); TA=TA+LOAASHO;    CALL ADDAXL(29,AL,STORE(I,1));    TBL06460
         TK=TK+LOADKY;    END; CALL MAXMINA(29); CALL MAXMINK(29);        TBL06470
END AXLEM6;                                                               TBL06480
EWLC : PROCEDURE;                                                         TBL12080
    DO I=1 TO 29;                                                         TBL12090
    IF MINLOADKY(I)=99999.0 THEN MINLOADKY(I)=0.0;                        TBL12100
    IF MINLOAASHO(I)=99999.0 THEN MINLOAASHO(I)=0.0;                      TBL12110
    N=AXLOVSAXAA(I,21) * (AXLOVSAXAA(I,21) - 1);                          TBL12120
    IF N=0 THEN DO; SDAASHO(I)=0; GO TO A1; END;                          TBL12130
    DWT = (AXLOVSAXAA(I,21)*SDAASHO(I) - MEANAASHO(I) ** 2) / N;          TBL12140
    IF DWT < 0 THEN DO; SDAASHO(I) = -1; GO TO A1; END;                   TBL12150
    SDAASHO(I)= SQRT(DWT);                                                TBL12160
A1: IF AXLOVSAXAA(I,21)=0 THEN DO; MEANAASHO(I)=0; GO TO A2; END;         TBL12170
    MEANAASHO(I) = MEANAASHO(I) / AXLOVSAXAA(I,21);                       TBL12180
A2: N=AXLOVSAX(I,21) * (AXLOVSAX(I,21) -1);                               TBL12190
    IF N=0 THEN DO; SDKY(I)=0; GO TO A3; END;                             TBL12200
    DWT = (AXLOVSAX(I,21) * SDKY(I) - MEANLOADKY(I) ** 2) / N;            TBL12210
    IF DWT < 0 THEN DO; SDKY(I)= -1; GO TO A3; END;                       TBL12220
    SDKY(I) = SQRT(DWT);                                                  TBL12230
A3: IF AXLOVSAX(I,21) = 0 THEN DO; MEANLOADKY(I)=0; GO TO A4; END;        TBL12240
    MEANLOADKY(I) = MEANLOADKY(I) / AXLOVSAX(I,21);                       TBL12250
A4: END;                                                                  TBL12260
    PUT FILE(SYSPRINT) PAGE LINE(4) EDIT ('EQUIVALENT AXLE LOAD PER',     TBL12270
         ' VEHICLE') (COLUMN(39),A,A);                                    TBL12280
    PUT FILE(SYSPRINT) SKIP(3) EDIT ('|  UNDER','AXLE PLACEMENT')        TBL12290
         (COLUMN(20),A,X(31),A);                                          TBL12300
    PUT FILE(SYSPRINT) SKIP EDIT ('    CATEGORIES     | 2 TONS') (A);     TBL12310
    PUT FILE(SYSPRINT)SKIP EDIT ('| 110      110      111      120 ',    TBL12320
         '     1111     1210     1120     1300    11111    12110')      TBL12330
```

```
     (COLUMN(20),A,A);                                                  TBL12340
PUT FILE(SYSPRINT) SKIP(0) EDIT (('__________' DO I=1 TO 11)) (A);      TBL12350
PUT FILE(SYSPRINT) SKIP EDIT ('| MAXIMUM  |',('        |' DO           TBL12360
     I=1  TO 10)) (COLUMN(9),A,10 A);                                   TBL12370
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MAXLOADKY(I),' |'        TBL12380
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12390
PUT FILE(SYSPRINT) SKIP EDIT('| MINIMUM  |',('        |' DO            TBL12400
     I=1 TO 10)) (COLUMN(9),A,10 A);                                    TBL12410
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MINLOADKY(I),' |'        TBL12420
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12430
PUT FILE(SYSPRINT) SKIP EDIT ('KENTUCKY|   MEAN   |',('        |'       TBL12440
     DO I=1 TO 10)) (A);                                                TBL12450
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MEANLOADKY(I),' |'       TBL12460
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12470
PUT FILE(SYSPRINT) SKIP EDIT ('| STANDARD |',('        |'               TBL12480
     DO I=1 TO 10)) (COLUMN(9),11 A);                                   TBL12490
PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION |',(SDKY(I),' |'              TBL12500
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12510
PUT FILE(SYSPRINT) SKIP EDIT ('|  TOTAL   |',('        |'               TBL12520
     DO I=1 TO 10))(COLUMN(9),11 A);                                    TBL12530
PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES |',(AXLDVSAX(I,21),' |'       TBL12540
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,0),A));                       TBL12550
PUT FILE(SYSPRINT) SKIP(0) EDIT (('__________' DO I=1 TO 11)) (A);      TBL12560
PUT FILE(SYSPRINT) SKIP EDIT ('| MAXIMUM  |',('        |' DO           TBL12570
     I=1  TO 10)) (COLUMN(9),A,10 A);                                   TBL12580
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MAXLDAASHO(I),' |'       TBL12590
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12600
PUT FILE(SYSPRINT) SKIP EDIT('| MINIMUM  |',('        |' DO            TBL12610
     I=1 TO 10)) (COLUMN(9),A,10 A);                                    TBL12620
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MINLDAASHO(I),' |'       TBL12630
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12640
PUT FILE(SYSPRINT) SKIP EDIT (' AASHO  |   MEAN   |',('        |'       TBL12650
     DO I=1 TO 10)) (A);                                                TBL12660
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(  MEANAASHO(I),' |'      TBL12670
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12680
PUT FILE(SYSPRINT) SKIP EDIT ('| STANDARD |',('        |'               TBL12690
     DO I=1 TO 10)) (COLUMN(9),11 A);                                   TBL12700
PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION |',(SDAASHO(I),' |'           TBL12710
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12720
PUT FILE(SYSPRINT) SKIP EDIT ('|  TOTAL   |',('        |'               TBL12730
     DO I=1 TO 10)) (COLUMN(9),11 A);                                   TBL12740
PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES |',(AXLDVSAXAA(I,21),' |'TBL12750
     DO I=1  TO 10)) (COLUMN(9),A,10 (F(7,0),A));                       TBL12760
PUT FILE(SYSPRINT) SKIP(0) EDIT (('__________' DO I=1 TO 11)) (A);      TBL12770
PUT FILE(SYSPRINT) SKIP(4) EDIT ('|        ','AXLE PLACEMENT')         TBL12780
     (COLUMN(20),A,X(31),A);                                            TBL12790
PUT FILE(SYSPRINT) SKIP EDIT ('    CATEGORIES     |        ') (A);       TBL12800
PUT FILE(SYSPRINT) SKIP EDIT ('| 11210    11120    13100    11300',TBL12810
     '    12200    111111   121110   112110   111210   111120')      TBL12820
     (COLUMN(20),A,A);                                                  TBL12830
PUT FILE(SYSPRINT) SKIP(0) EDIT (('__________' DO I=1 TO 11)) (A);      TBL12840
PUT FILE(SYSPRINT) SKIP EDIT ('| MAXIMUM  |',('        |' DO           TBL12850
     I=1  TO 10)) (COLUMN(9),A,10 A);                                   TBL12860
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MAXLOADKY(I),' |'        TBL12870
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12880
PUT FILE(SYSPRINT) SKIP EDIT('| MINIMUM  |',('        |' DO            TBL12890
     I=1 TO 10)) (COLUMN(9),A,10 A);                                    TBL12900
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MINLOADKY(I),' |'        TBL12910
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                       TBL12920
PUT FILE(SYSPRINT) SKIP EDIT ('KENTUCKY|   MEAN   |',('        |'       TBL12930
     DO I=1 TO 10)) (A);                                                TBL12940
```

```
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(MEANLOADKY(I),' |'   TBL12950
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                   TBL12960
PUT FILE(SYSPRINT) SKIP EDIT ('| STANDARD |',('         |'          TBL12970
     DO I=1 TO 10)) (COLUMN(9),11 A);                               TBL12980
PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION |',(SDKY(I),' |'         TBL12990
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13000
PUT FILE(SYSPRINT) SKIP EDIT ('|  TOTAL   |',('         |'          TBL13010
     DO I=1 TO 10))(COLUMN(9),11 A);                                TBL13020
PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES |',(AXLDVSAX(I,21),' |'  TBL13030
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,0),A));                   TBL13040
PUT FILE(SYSPRINT) SKIP(0) EDIT (('__________' DO I=1 TO 11)) (A); TBL13050
PUT FILE(SYSPRINT) SKIP EDIT ('| MAXIMUM  |',('         |' DO       TBL13060
     I=1  TO 10)) (COLUMN(9),A,10 A);                               TBL13070
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(MAXLDAASHO(I),' |'  TBL13080
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13090
PUT FILE(SYSPRINT) SKIP EDIT('| MINIMUM  |',('         |' DO        TBL13100
     I=1 TO 10)) (COLUMN(9),A,10 A);                                TBL13110
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(MINLDAASHO(I),' |'  TBL13120
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13130
PUT FILE(SYSPRINT) SKIP EDIT (' AASHO  |   MEAN   |',('         |'  TBL13140
     DO I=1 TO 10)) (A);                                            TBL13150
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(  MEANAASHO(I),' |'  TBL13160
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13170
PUT FILE(SYSPRINT) SKIP EDIT ('| STANDARD |',('         |'          TBL13180
     DO I=1 TO 10)) (COLUMN(9),11 A);                               TBL13190
PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION |',(SDAASHO(I),' |'      TBL13200
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13210
PUT FILE(SYSPRINT) SKIP EDIT ('|  TOTAL   |',('         |'          TBL13220
     DO I=1 TO 10)) (COLUMN(9),11 A);                               TBL13230
PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES |',(AXLDVSAXAA(I,21),' |'TBL13240
     DO I=11 TO 20)) (COLUMN(9),A,10 (F(7,0),A));                   TBL13250
PUT FILE(SYSPRINT) SKIP(0) EDIT (('__________' DO I=1 TO 11)) (A); TBL13260
PUT FILE(SYSPRINT)  PAGE;                                           TBL13270
PUT FILE(SYSPRINT) SKIP(4) EDIT ('|       ','AXLE PLACEMENT')      TBL13280
     (COLUMN(20),A,X(26),A);                                        TBL13290
PUT FILE(SYSPRINT) SKIP EDIT ('    CATEGORIES     |       ') (A);  TBL13300
PUT FILE(SYSPRINT) SKIP EDIT ('| 122100   112200   121200   13200',TBL13310
     '0   123000   131100   113100   11130   OVER 6')              TBL13320
     (COLUMN(20),A,A);                                              TBL13330
PUT FILE(SYSPRINT) SKIP(0) EDIT (('_' DO I=1 TO 101)) (A);         TBL13340
PUT FILE(SYSPRINT) SKIP EDIT ('| MAXIMUM  |',('         |' DO       TBL13350
     I=1  TO 9 )) (COLUMN(9),A,10 A);                               TBL13360
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(MAXLOADKY(I),' |'   TBL13370
     DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13380
PUT FILE(SYSPRINT) SKIP EDIT('| MINIMUM  |',('         |' DO        TBL13390
     I=1 TO 9 )) (COLUMN(9),A,10 A);                                TBL13400
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(MINLOADKY(I),' |'   TBL13410
     DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13420
PUT FILE(SYSPRINT) SKIP EDIT ('KENTUCKY|   MEAN   |',('         |'  TBL13430
     DO I=1 TO 9 )) (A);                                            TBL13440
PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL     |',(MEANLOADKY(I),' |'   TBL13450
     DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13460
PUT FILE(SYSPRINT) SKIP EDIT ('| STANDARD |',('         |'          TBL13470
     DO I=1 TO 9 )) (COLUMN(9),11 A);                               TBL13480
PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION |',(SDKY(I),' |'         TBL13490
     DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                   TBL13500
PUT FILE(SYSPRINT) SKIP EDIT ('|  TOTAL   |',('         |'          TBL13510
     DO I=1 TO 9 ))(COLUMN(9),11 A);                                TBL13520
PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES |',(AXLDVSAX(I,21),' |'  TBL13530
     DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,0),A));                   TBL13540
PUT FILE(SYSPRINT) SKIP(0) EDIT (('_' DO I=1 TO 101)) (A);         TBL13550
```

```
PUT FILE(SYSPRINT) SKIP EDIT ('(AASHO CATEGORIES)')(COLUMN(48),A); TBL07390
PUT FILE(SYSPRINT) SKIP(2) EDIT (' TANDEM SPACING IS 40 INCHES ', TBL07400
     'OR LESS ')(COLUMN(38),A,A); TBL07410
PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE,BITAN',TBL07420
     'DEM AND TRITANDEM AXLES') (COLUMN(31),A,A); TBL07430
PUT FILE(SYSPRINT) SKIP(5) LIST (' GROSS |'); TBL07440
PUT FILE(SYSPRINT) SKIP EDIT ('OPERATING | UNDER','AXLE PLACEMEN',TBL07450
     'T ') (A,X(35),A); TBL07460
PUT FILE(SYSPRINT) SKIP LIST (' WEIGHT | 2 TONS'); TBL07470
PUT FILE(SYSPRINT) SKIP EDIT(' (KIPS) | 110 110 111 1',TBL07480
     '20 1111 1210 1120 1300 11111 12110 11210 11120',TBL07490
     ' 13100 11300 12200 ') (A,A,A); TBL07500
PUT FILE(SYSPRINT) SKIP(-1); TBL07510
DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT ('____') (A); END; TBL07520
PUT FILE(SYSPRINT) SKIP LIST (' UNDER 4 |'); CALL PRGWAX1(1); TBL07530
PUT FILE(SYSPRINT) SKIP LIST (' 4 - 10 |'); CALL PRGWAX1(2); TBL07540
PUT FILE(SYSPRINT) SKIP LIST (' 10 - 15 |'); CALL PRGWAX1(3); TBL07550
PUT FILE(SYSPRINT) SKIP LIST (' 15 - 20 |'); CALL PRGWAX1(4); TBL07560
PUT FILE(SYSPRINT) SKIP LIST (' 20 - 22 |'); CALL PRGWAX1(5); TBL07570
PUT FILE(SYSPRINT) SKIP LIST (' 22 - 24 |'); CALL PRGWAX1(6); TBL07580
PUT FILE(SYSPRINT) SKIP LIST (' 24 - 26 |'); CALL PRGWAX1(7); TBL07590
PUT FILE(SYSPRINT) SKIP LIST (' 26 - 28 |'); CALL PRGWAX1(8); TBL07600
PUT FILE(SYSPRINT) SKIP LIST (' 28 - 30 |'); CALL PRGWAX1(9); TBL07610
PUT FILE(SYSPRINT) SKIP LIST (' 30 - 32 |'); CALL PRGWAX1(10); TBL07620
PUT FILE(SYSPRINT) SKIP LIST (' 32 - 34 |'); CALL PRGWAX1(11); TBL07630
PUT FILE(SYSPRINT) SKIP LIST (' 34 - 36 |'); CALL PRGWAX1(12); TBL07640
PUT FILE(SYSPRINT) SKIP LIST (' 36 - 38 |'); CALL PRGWAX1(13); TBL07650
PUT FILE(SYSPRINT) SKIP LIST (' 38 - 40 |'); CALL PRGWAX1(14); TBL07660
PUT FILE(SYSPRINT) SKIP LIST (' 40 - 45 |'); CALL PRGWAX1(15); TBL07670
PUT FILE(SYSPRINT) SKIP LIST (' 45 - 50 |'); CALL PRGWAX1(16); TBL07680
PUT FILE(SYSPRINT) SKIP LIST (' 50 - 55 |'); CALL PRGWAX1(17); TBL07690
PUT FILE(SYSPRINT) SKIP LIST (' 55 - 60 |'); CALL PRGWAX1(18); TBL07700
PUT FILE(SYSPRINT) SKIP LIST (' 60 - 65 |'); CALL PRGWAX1(19); TBL07710
PUT FILE(SYSPRINT) SKIP LIST (' 65 - 60 |'); CALL PRGWAX1(20); TBL07720
PUT FILE(SYSPRINT) SKIP LIST (' 70 - 75 |'); CALL PRGWAX1(21); TBL07730
PUT FILE(SYSPRINT) SKIP LIST (' 75 - 80 |'); CALL PRGWAX1(22); TBL07740
PUT FILE(SYSPRINT) SKIP LIST (' 80 - 85 |'); CALL PRGWAX1(23); TBL07750
PUT FILE(SYSPRINT) SKIP LIST (' 85 - 90 |'); CALL PRGWAX1(24); TBL07760
PUT FILE(SYSPRINT) SKIP LIST (' 90 - 95 |'); CALL PRGWAX1(25); TBL07770
PUT FILE(SYSPRINT) SKIP LIST (' OVER 95 |'); CALL PRGWAX1(26); TBL07780
PUT FILE(SYSPRINT) SKIP(1) LIST (' TOTAL |'); TBL07790
DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT(' |')(A); END; TBL07800
PUT FILE(SYSPRINT) SKIP LIST (' VEHICLES |'); CALL PRGWAX1(27); TBL07810
PUT FILE(SYSPRINT) SKIP LIST ('MEAN GROSS|'); TBL07820
DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT(' |')(A); END; TBL07830
PUT FILE(SYSPRINT) SKIP LIST (' WEIGHT |'); TBL07840
DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (SDGOWAX(1,I),' |')(F(5,1), TBL07850
     A); END; TBL07860
PUT FILE(SYSPRINT) SKIP LIST (' STANDARD |'); TBL07870
DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT(' |')(A); END; TBL07880
PUT FILE(SYSPRINT) SKIP LIST (' DEVIATION|'); TBL07890
DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT (SDGOWAX(2,I),' |')(FI5,1), TBL07900
     A); END ;PUT FILE(SYSPRINT) SKIP(0); TBL07910
DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT ('____') (A); END; TBL07920
PUT FILE(SYSPRINT) PAGE LINE (6) EDIT ('GROSS OPERATING WEIGHT ', TBL07930
     'VERSUS AXLE PLACEMENT (CONTINUED FROM PRECEDING PAGE)') TBL07940
     (COLUMN(25),A,A); TBL07950
PUT FILE(SYSPRINT) SKIP(2) EDIT (' TANDEM SPACING IS 40 INCHES ', TBL07960
     'OR LESS ')(COLUMN(38),A,A); TBL07970
PUT FILE(SYSPRINT) SKIP EDIT ('(AASHO CATEGORIES)')(COLUMN(48),A); TBL07980
PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE,BITAN',TBL07990
```

```
     PUT FILE(SYSPRINT) SKIP LIST (' 28 - 30  |'); CALL PRGWOS(9);      TBL06830
     PUT FILE(SYSPRINT) SKIP LIST (' 30 - 32  |'); CALL PRGWOS(10);     TBL06840
     PUT FILE(SYSPRINT) SKIP LIST (' 32 - 34  |'); CALL PRGWOS(11);     TBL06850
     PUT FILE(SYSPRINT) SKIP LIST (' 34 - 36  |'); CALL PRGWOS(12);     TBL06860
     PUT FILE(SYSPRINT) SKIP LIST (' 36 - 38  |'); CALL PRGWOS(13);     TBL06870
     PUT FILE(SYSPRINT) SKIP LIST (' 38 - 40  |'); CALL PRGWOS(14);     TBL06880
     PUT FILE(SYSPRINT) SKIP LIST (' 40 - 45  |'); CALL PRGWOS(15);     TBL06890
     PUT FILE(SYSPRINT) SKIP LIST (' 45 - 50  |'); CALL PRGWOS(16);     TBL06900
     PUT FILE(SYSPRINT) SKIP LIST (' 50 - 55  |'); CALL PRGWOS(17);     TBL06910
     PUT FILE(SYSPRINT) SKIP LIST (' 55 - 60  |'); CALL PRGWOS(18);     TBL06920
     PUT FILE(SYSPRINT) SKIP LIST (' 60 - 65  |'); CALL PRGWOS(19);     TBL06930
     PUT FILE(SYSPRINT) SKIP LIST (' 65 - 70  |'); CALL PRGWOS(20);     TBL06940
     PUT FILE(SYSPRINT) SKIP LIST (' 70 - 75  |'); CALL PRGWOS(21);     TBL06950
     PUT FILE(SYSPRINT) SKIP LIST (' 75 - 80  |'); CALL PRGWOS(22);     TBL06960
     PUT FILE(SYSPRINT) SKIP LIST (' 80 - 85  |'); CALL PRGWOS(23);     TBL06970
     PUT FILE(SYSPRINT) SKIP LIST (' 85 - 90  |'); CALL PRGWOS(24);     TBL06980
     PUT FILE(SYSPRINT) SKIP LIST (' 90 - 95  |'); CALL PRGWOS(25);     TBL06990
     PUT FILE(SYSPRINT) SKIP LIST (' OVER 95  |'); CALL PRGWOS(26);     TBL07000
     PUT FILE(SYSPRINT) SKIP    LIST ('  TOTAL   |');                   TBL07010
     DO I=1 TO 11; PUT FILE(SYSPRINT) EDIT('      |')(A); END;          TBL07020
     PUT FILE(SYSPRINT) SKIP    LIST (' VEHICLES |'); CALL PRGWOS(27); TBL07030
     PUT FILE(SYSPRINT) SKIP    LIST ('MEAN GROSS|');                 TBL07040
     DO I=1 TO 11; PUT FILE(SYSPRINT) EDIT('      |')(A); END;          TBL07050
     PUT FILE(SYSPRINT) SKIP    LIST ('  WEIGHT  |');                 TBL07060
     DO I=1 TO 10;  PUT FILE(SYSPRINT) EDIT (SDGOWOS(1,I),' |')(F(5,1), TBL07070
          A); END;                                                     TBL07080
     PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                          TBL07090
     PUT FILE(SYSPRINT) SKIP    LIST (' STANDARD |');                 TBL07100
     DO I=1 TO 11; PUT FILE(SYSPRINT) EDIT('      |')(A); END;          TBL07110
     PUT FILE(SYSPRINT) SKIP    LIST(' DEVIATION|');                  TBL07120
     DO I=1 TO 10;  PUT FILE(SYSPRINT) EDIT (SDGOWOS(2,I),' |')       TBL07130
                  (F(5,1),A);  END;                                    TBL07140
     PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                          TBL07150
     PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL07160
          '_____________________________________________') (A,A);      TBL07170
                                                                       TBL07180
 PRGWOS : PROCEDURE(CA);                                               TBL07200
     DCL CA FIXED BINARY (31);                                         TBL07210
     DO I=1 TO 11;                                                     TBL07220
     PUT FILE(SYSPRINT) EDIT (GOWVSOS(I,CA),' |')(F(5,0),A);           TBL07230
     END;                                                              TBL07240
 END PRGWOS;                                                           TBL07250
 END GWVSOSC;                                                          TBL07190
* PROCESS ('OPT=1')
 GWAXC : PROCEDURE (GOWVSAX,SDGOWAX);                                  TBL07260
     DECLARE   PRGWAX1 ENTRY (FIXED BIN (31)),                         TBL00430
               PRGWAX2 ENTRY (FIXED BIN (31));                         TBL00440
     DCL  GOWVSAX(30,27) FIXED BIN(31),                                TBL07261
          (SDGOWAX(3,30),DWT) FLOAT BIN (16);                          TBL07262
     DO I=1 TO 29; N=GOWVSAX(I,27) * (GOWVSAX(I,27) - 1);              TBL07270
     IF N=0 THEN DO;SDGOWAX(2,I) =0;  GO TO G1; END;                   TBL07280
     DWT = (GOWVSAX(I,27) * SDGOWAX(2,I) - SDGOWAX(1,I) ** 2) / N;     TBL07290
     IF DWT < 0 THEN DO; SDGOWAX(2,I) = -1; GO TO G1; END;             TBL07300
     SDGOWAX(2,I) = SQRT (DWT);                                        TBL07310
 G1: IF GOWVSAX(I,27)=0 THEN DO ;SDGOWAX(1,I)= 0;  GO TO G2; END;      TBL07320
     SDGOWAX(1,I) = SDGOWAX(1,I) / GOWVSAX(I,27);                      TBL07330
     GOWVSAX(30,27) = GOWVSAX(30,27) + GOWVSAX(I,27);                  TBL07340
 G2: END;                                                              TBL07350
     PUT FILE(SYSPRINT) PAGE;                                          TBL07360
     PUT FILE(SYSPRINT) SKIP(5) EDIT ('GROSS OPERATING WEIGHT VERSUS ', TBL07370
          'AXLE PLACEMENT')(COLUMN(35),A,A);                           TBL07380
```

```
     PUT FILE(SYSPRINT) SKIP EDIT ('| MAXIMUM  |',('        |' DO        TBL13560
          I=1  TO 9 )) (COLUMN(9),A,10 A);                                TBL13570
     PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MAXLDAASHO(I),' |'   TBL13580
          DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                    TBL13590
     PUT FILE(SYSPRINT) SKIP EDIT('| MINIMUM  |',('        |' DO        TBL13600
          I=1 TO 9 )) (COLUMN(9),A,10 A);                                 TBL13610
     PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(MINLDAASHO(I),' |'   TBL13620
          DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                    TBL13630
     PUT FILE(SYSPRINT) SKIP EDIT (' AASHO  |   MEAN   |',('        |'   TBL13640
          DO I=1 TO 9 )) (A);                                             TBL13650
     PUT FILE(SYSPRINT) SKIP EDIT ('|   EAL    |',(  MEANAASHO(I),' |'  TBL13660
          DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                    TBL13670
     PUT FILE(SYSPRINT) SKIP EDIT ('| STANDARD |',('        |'           TBL13680
          DO I=1 TO  9)) (COLUMN(9),11 A);                                TBL13690
     PUT FILE(SYSPRINT) SKIP EDIT ('|DEVIATION |',(SDAASHO(I),' |'       TBL13700
          DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,1),A));                    TBL13710
     PUT FILE(SYSPRINT) SKIP EDIT ('|  TOTAL   |',('        |'           TBL13720
          DO I=1 TO  9)) (COLUMN(9),11 A);                                TBL13730
     PUT FILE(SYSPRINT) SKIP EDIT ('| VEHICLES |',(AXLOVSAXAA(I,21),' |'TBL13740
          DO I=21 TO 29)) (COLUMN(9),A,10 (F(7,0),A));                    TBL13750
     PUT FILE(SYSPRINT) SKIP(0) EDIT (('_' DO I=1 TO 101)) (A);           TBL13760
 END EWLC;                                                                TBL13770
 DONE : END VEHDATA;                                                      TBL13780
* PROCESS ('OPT=1')
 GWVSOSC : PROCEDURE (GOWVSOS,SDGOWOS);                                   TBL06490
     DECLARE   PRGWOS ENTRY (FIXED BIN (31));                             TBL00420
     DCL    GOWVSOS(11,27) FIXED BIN(31),                                 TBL06491
          (SDGOWOS(2,10),DWT) FLOAT BIN (16);                             TBL06492
     DO I=1 TO 10;                                                        TBL06500
     IF (GOWVSOS(I,27) * (GOWVSOS(I,27) -1)) = 0 THEN DO; SDGOWOS(2,I)    TBL06510
          =0; GO TO G1; END;                                              TBL06520
     DWT = (GOWVSOS(I,27) * SDGOWOS(2,I) - SDGOWOS(1,I) ** 2)/            TBL06530
          (GOWVSOS(I,27) * (GOWVSOS(I,27) - 1));                          TBL06540
     IF DWT < 0.0 THEN DO; SDGOWOS(2,I) = -1; GO TO G1; END;              TBL06550
     SDGOWOS(2,I) = SQRT(DWT);                                            TBL06560
 G1: IF GOWVSOS(I,27) = 0 THEN DO; SDGOWOS(1,I) = 0; GO TO G2; END;       TBL06570
     SDGOWOS(1,I) = SDGOWOS(1,I) / GOWVSOS(I,27);                         TBL06580
     GOWVSOS(11,27)= GOWVSOS(11,27) + GOWVSOS(I,27);                      TBL06590
 G2: END;                                                                 TBL06600
     PUT FILE(SYSPRINT) PAGE;                                             TBL06610
     PUT FILE(SYSPRINT) SKIP (4) EDIT ('GROSS OPERATING WEIGHT VERSUS',   TBL06620
          ' OPERATING SPEED') (COLUMN(17),A,A);                           TBL06630
     PUT FILE(SYSPRINT) SKIP (5) LIST ('  GROSS   |');                   TBL06640
     PUT FILE(SYSPRINT) SKIP     EDIT ('OPERATING |','SPEED (MPH)')       TBL06650
          (A,X(31),A);                                                    TBL06660
     PUT FILE(SYSPRINT) SKIP    EDIT ('  WEIGHT  | UNDER',                TBL06670
          'OVER   TOTAL') (A,X(59),A);                                    TBL06680
     PUT FILE(SYSPRINT) SKIP EDIT('  (KIPS)  |   20   20-40  40-50  ',    TBL06690
          '50-55  55-60  60-65  65-70  70-80  80-90    90  VEHICLES')     TBL06700
          (A,A);                                                          TBL06710
     PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',    TBL06720
          '_____________________________________________________')        TBL06730
          (A,A);                                                          TBL06740
     PUT FILE(SYSPRINT) SKIP LIST (' UNDER 4  |'); CALL PRGWOS(1);        TBL06750
     PUT FILE(SYSPRINT) SKIP LIST ('  4 - 10  |'); CALL PRGWOS(2);        TBL06760
     PUT FILE(SYSPRINT) SKIP LIST (' 10 - 15  |'); CALL PRGWOS(3);        TBL06770
     PUT FILE(SYSPRINT) SKIP LIST (' 15 - 20  |'); CALL PRGWOS(4);        TBL06780
     PUT FILE(SYSPRINT) SKIP LIST (' 20 - 22  |'); CALL PRGWOS(5);        TBL06790
     PUT FILE(SYSPRINT) SKIP LIST (' 22 - 24  |'); CALL PRGWOS(6);        TBL06800
     PUT FILE(SYSPRINT) SKIP LIST (' 24 - 26  |'); CALL PRGWOS(7);        TBL06810
     PUT FILE(SYSPRINT) SKIP LIST (' 26 - 28  |'); CALL PRGWOS(8);        TBL06820
```

```
     'DEM AND TRITANDEM AXLES')  (COLUMN(31),A,A);                    TBL08000
    PUT FILE(SYSPRINT) SKIP(5) LIST ('  GROSS    |');                  TBL08010
    PUT FILE(SYSPRINT) SKIP EDIT ('OPERATING |       ','AXLE PLACEMEN',TBL08020
     'T       ') (A,X(35),A);                                         TBL08030
    PUT FILE(SYSPRINT) SKIP EDIT ('  WEIGHT  |       ','TOTAL')       TBL08040
     (A,X(93),A);                                                    TBL08050
    PUT FILE(SYSPRINT) SKIP EDIT ('  (KIPS)  |111111 121110 112110 11',TBL08060
     '1210 111120 122100 112200 121200 132000 123000 131100 11310',TBL08070
     '0 111300 OVER 6 VEHICLES') (A,A,A);                            TBL08080
     PUT FILE(SYSPRINT) SKIP(0);                                     TBL08090
    DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT ('____') (A); END;         TBL08100
    PUT FILE(SYSPRINT) SKIP LIST (' UNDER 4  |'); CALL PRGWAX2(1);   TBL08110
    PUT FILE(SYSPRINT) SKIP LIST ('  4 - 10  |'); CALL PRGWAX2(2);   TBL08120
    PUT FILE(SYSPRINT) SKIP LIST (' 10 - 15  |'); CALL PRGWAX2(3);   TBL08130
    PUT FILE(SYSPRINT) SKIP LIST (' 15 - 20  |'); CALL PRGWAX2(4);   TBL08140
    PUT FILE(SYSPRINT) SKIP LIST (' 20 - 22  |'); CALL PRGWAX2(5);   TBL08150
    PUT FILE(SYSPRINT) SKIP LIST (' 22 - 24  |'); CALL PRGWAX2(6);   TBL08160
    PUT FILE(SYSPRINT) SKIP LIST (' 24 - 26  |'); CALL PRGWAX2(7);   TBL08170
    PUT FILE(SYSPRINT) SKIP LIST (' 26 - 28  |'); CALL PRGWAX2(8);   TBL08180
    PUT FILE(SYSPRINT) SKIP LIST (' 28 - 30  |'); CALL PRGWAX2(9);   TBL08190
    PUT FILE(SYSPRINT) SKIP LIST (' 30 - 32  |'); CALL PRGWAX2(10);  TBL08200
    PUT FILE(SYSPRINT) SKIP LIST (' 32 - 34  |'); CALL PRGWAX2(11);  TBL08210
    PUT FILE(SYSPRINT) SKIP LIST (' 34 - 36  |'); CALL PRGWAX2(12);  TBL08220
    PUT FILE(SYSPRINT) SKIP LIST (' 36 - 38  |'); CALL PRGWAX2(13);  TBL08230
    PUT FILE(SYSPRINT) SKIP LIST (' 38 - 40  |'); CALL PRGWAX2(14);  TBL08240
    PUT FILE(SYSPRINT) SKIP LIST (' 40 - 45  |'); CALL PRGWAX2(15);  TBL08250
    PUT FILE(SYSPRINT) SKIP LIST (' 45 - 50  |'); CALL PRGWAX2(16);  TBL08260
    PUT FILE(SYSPRINT) SKIP LIST (' 50 - 55  |'); CALL PRGWAX2(17);  TBL08270
    PUT FILE(SYSPRINT) SKIP LIST (' 55 - 60  |'); CALL PRGWAX2(18);  TBL08280
    PUT FILE(SYSPRINT) SKIP LIST (' 60 - 65  |'); CALL PRGWAX2(19);  TBL08290
    PUT FILE(SYSPRINT) SKIP LIST (' 65 - 70  |'); CALL PRGWAX2(20);  TBL08300
    PUT FILE(SYSPRINT) SKIP LIST (' 70 - 75  |'); CALL PRGWAX2(21);  TBL08310
    PUT FILE(SYSPRINT) SKIP LIST (' 75 - 80  |'); CALL PRGWAX2(22);  TBL08320
    PUT FILE(SYSPRINT) SKIP LIST (' 80 - 85  |'); CALL PRGWAX2(23);  TBL08330
    PUT FILE(SYSPRINT) SKIP LIST (' 85 - 90  |'); CALL PRGWAX2(24);  TBL08340
    PUT FILE(SYSPRINT) SKIP LIST (' 90 - 95  |'); CALL PRGWAX2(25);  TBL08350
    PUT FILE(SYSPRINT) SKIP LIST (' OVER 95  |'); CALL PRGWAX2(26);  TBL08360
    PUT FILE(SYSPRINT) SKIP    LIST ('  TOTAL    |');                TBL08370
    DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT('       |')(A); END;       TBL08380
    PUT FILE(SYSPRINT) SKIP    LIST (' VEHICLES |'); CALL PRGWAX2(27); TBL08390
    PUT FILE(SYSPRINT) SKIP    LIST ('MEAN GROSS|');                TBL08400
    DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT('       |')(A); END;       TBL08410
    PUT FILE(SYSPRINT) SKIP    LIST ('  WEIGHT  |');                TBL08420
    DO I=16 TO 29; PUT FILE(SYSPRINT) EDIT (SDGOWAX(1,I),' |')(F(5,1), TBL08430
     A); END;                                                        TBL08440
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                         TBL08450
    PUT FILE(SYSPRINT) SKIP    LIST (' STANDARD |');                 TBL08460
    DO I=1 TO 15; PUT FILE(SYSPRINT) EDIT('       |')(A); END;       TBL08470
    PUT FILE(SYSPRINT) SKIP    LIST (' DEVIATION|');                 TBL08480
    DO I=16 TO 29; PUT FILE(SYSPRINT) EDIT (SDGOWAX(2,I),' |')(F(5,1), TBL08490
     A); END;                                                        TBL08500
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                         TBL08510
    PUT FILE(SYSPRINT) SKIP(0);                                      TBL08520
    DO I=1 TO 29; PUT FILE(SYSPRINT) EDIT ('____') (A); END;         TBL08530
PRGWAX1 : PROCEDURE (CA);                                            TBL08550
    DCL CA FIXED BINARY (31);                                        TBL08560
    DO I=1 TO 15;                                                    TBL08570
    PUT FILE(SYSPRINT) EDIT (GOWVSAX(I,CA),' |') (F(5,0),A); END;    TBL08580
END PRGWAX1;                                                         TBL08590
PRGWAX2 : PROCEDURE (CA);                                            TBL08600
    DCL CA FIXED BINARY (31);                                        TBL08610
```

```
     DO I=16 TO 30;                                                        TBL08620
     PUT FILE(SYSPRINT) EDIT (GOWVSAX(I,CA),' |') (F(5,0),A); END;         TBL08630
 END PRGWAX2;                                                              TBL08640
 END GWAXC;                                                                TBL08540
* PROCESS ('OPT=1')
 OSVSAXC : PROCEDURE (OSVSAX,SDOSAX);                                      TBL08650
     DECLARE    PROSAX1 ENTRY (FIXED BIN (31)),                            TBL00490
                PROSAX2 ENTRY (FIXED BIN (31));                            TBL00500
     DCL OSVSAX(30,11) FIXED BIN (31),
         (SDOSAX(3,30),DWT) FLOAT BIN(16);
     DO I=1 TO 29;                                                         TBL08660
     N = OSVSAX(I,11) * (OSVSAX(I,11) - 1);                                TBL08670
     IF N=0 THEN DO; SDOSAX(2,I) = 0; GO TO G1; END;                       TBL08680
     DWT = (OSVSAX(I,11) * SDOSAX(2,I) - SDOSAX(1,I) ** 2) / N;            TBL08690
     IF DWT<0 THEN DO; SDOSAX(2,I) = -1; GO TO G1; END;                    TBL08700
     SDOSAX(2,I) = SQRT(DWT);                                              TBL08710
 G1: IF OSVSAX(I,11) = 0 THEN DO; SDOSAX(1,I) = 0; GO TO G2; END;          TBL08720
     SDOSAX(1,I) = SDOSAX(1,I) / OSVSAX(I,11);                             TBL08730
     OSVSAX(30,11) = OSVSAX(30,11) + OSVSAX(I,11);                         TBL08740
 G2: END;                                                                  TBL08750
     PUT FILE(SYSPRINT) PAGE LINE(4)  EDIT ('OPERATING SPEED VERSUS AX',TBL08760
         'LE PLACEMENT','TANDEM SPACING IS 40 INCHES OR LESS','(AASHO',TBL08770
         ' CATEGORIES)','1,2 AND 3 INDICATE SINGLE,BITANDEM AND TRITA',TBL08780
         'NDEM AXLES') (COLUMN(40),A,A,SKIP(2),COLUMN(39),A,COLUMN(49),TBL08790
         A,A,SKIP(2),COLUMN(31),A,A);                                      TBL08800
     PUT FILE(SYSPRINT) SKIP(5)  EDIT ('AXLE PLACEMENT')(COLUMN(53),A);    TBL08810
     PUT FILE(SYSPRINT) SKIP EDIT ('OPERATING | UNDER',' SPEED   | 2', TBL08820
         ' TONS',' (MPH)   |  110    110    111    120    1111   1210',TBL08830
         '   1120   1300  11111  12110  11210  11120  13100  11300  1',TBL08840
         '2200') (A,SKIP,A,A,SKIP,A,A,A);                                  TBL08850
     PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL08860
         '_____________________________________________________',TBL08870
         '_____________________') (A,A,A);                                 TBL08880
     PUT FILE(SYSPRINT) SKIP EDIT ('UNDER 20  |')(A); CALL PROSAX1(1);     TBL08890
     PUT FILE(SYSPRINT) SKIP EDIT (' 20 - 40  |')(A); CALL PROSAX1(2);     TBL08900
     PUT FILE(SYSPRINT) SKIP EDIT (' 40 - 50  |')(A); CALL PROSAX1(3);     TBL08910
     PUT FILE(SYSPRINT) SKIP EDIT (' 50 - 55  |')(A); CALL PROSAX1(4);     TBL08920
     PUT FILE(SYSPRINT) SKIP EDIT (' 55 - 60  |')(A); CALL PROSAX1(5);     TBL08930
     PUT FILE(SYSPRINT) SKIP EDIT (' 60 - 65  |')(A); CALL PROSAX1(6);     TBL08940
     PUT FILE(SYSPRINT) SKIP EDIT (' 65 - 70  |')(A); CALL PROSAX1(7);     TBL08950
     PUT FILE(SYSPRINT) SKIP EDIT (' 70 - 80  |')(A); CALL PROSAX1(8);     TBL08960
     PUT FILE(SYSPRINT) SKIP EDIT (' 80 - 90  |')(A); CALL PROSAX1(9);     TBL08970
     PUT FILE(SYSPRINT) SKIP EDIT (' OVER 90  |')(A); CALL PROSAX1(10);    TBL08980
     PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |') (A);                     TBL08990
     DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT('      |') (A); END;             TBL09000
     PUT FILE(SYSPRINT) SKIP EDIT (' VEHICLES |')(A); CALL PROSAX1(11);    TBL09010
     PUT FILE(SYSPRINT) SKIP EDIT ('   MEAN   |') (A);                     TBL09020
     DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT('      |') (A); END;             TBL09030
     PUT FILE(SYSPRINT) SKIP EDIT ('  SPEED   |')(A);                      TBL09040
     DO I= 1 TO 15;PUT FILE(SYSPRINT) EDIT (SDOSAX(1,I),' |')(F(5,1),A);TBL09050
         END;                                                              TBL09060
     PUT FILE(SYSPRINT) SKIP EDIT (' STANDARD |')(A);                      TBL09070
     DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT('      |') (A); END;             TBL09080
     PUT FILE(SYSPRINT) SKIP EDIT('DEVIATION |') (A);                      TBL09090
     DO I=1 TO  15; PUT FILE(SYSPRINT) EDIT (SDOSAX(2,I),' |')(F(5,1),  TBL09100
         A); END;                                                          TBL09110
     PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL09120
         '_____________________________________________________',TBL09130
         '_____________________') (A,A,A);                                 TBL09140
     PUT FILE(SYSPRINT) SKIP(5)  EDIT ('AXLE PLACEMENT')(COLUMN(53),A); TBL09150
     PUT FILE(SYSPRINT) SKIP EDIT ('OPERATING |',' SPEED   |',          TBL09160
```

```
        ' TOTAL',' (MPH)   |111111 121110 112110 111210 111120 1221',TBL09170
        '00 112200 121200 132000 123000 131100 113100 111300 OVER 6 ',TBL09180
        'VEHICLES')   (A,SKIP,A,X(98),A,SKIP,A,A,A);                 TBL09190
    PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL09200
        '_____________________________________________________',TBL09210
        '_________________________') (A,A,A);                        TBL09220
    PUT FILE(SYSPRINT) SKIP EDIT ('UNDER 20  |')(A); CALL PROSAX2(1); TBL09230
    PUT FILE(SYSPRINT) SKIP EDIT (' 20 - 40  |')(A); CALL PROSAX2(2); TBL09240
    PUT FILE(SYSPRINT) SKIP EDIT (' 40 - 50  |')(A); CALL PROSAX2(3); TBL09250
    PUT FILE(SYSPRINT) SKIP EDIT (' 50 - 55  |')(A); CALL PROSAX2(4); TBL09260
    PUT FILE(SYSPRINT) SKIP EDIT (' 55 - 60  |')(A); CALL PROSAX2(5); TBL09270
    PUT FILE(SYSPRINT) SKIP EDIT (' 60 - 65  |')(A); CALL PROSAX2(6); TBL09280
    PUT FILE(SYSPRINT) SKIP EDIT (' 65 - 70  |')(A); CALL PROSAX2(7); TBL09290
    PUT FILE(SYSPRINT) SKIP EDIT (' 70 - 80  |')(A); CALL PROSAX2(8); TBL09300
    PUT FILE(SYSPRINT) SKIP EDIT (' 80 - 90  |')(A); CALL PROSAX2(9); TBL09310
    PUT FILE(SYSPRINT) SKIP EDIT (' OVER 90  |')(A); CALL PROSAX2(10); TBL09320
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |') (A);                TBL09330
    DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT('      |') (A); END;         TBL09340
    PUT FILE(SYSPRINT) SKIP EDIT (' VEHICLES |')(A); CALL PROSAX2(11); TBL09350
    PUT FILE(SYSPRINT) SKIP EDIT ('   MEAN   |') (A);                TBL09360
    DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT('      |') (A); END;         TBL09370
    PUT FILE(SYSPRINT) SKIP EDIT ('  SPEED   |')(A);                 TBL09380
    DO I=16 TO 29;PUT FILE(SYSPRINT) EDIT (SDOSAX(1,I),' |')(F(5,1),A);TBL09390
        END;                                                         TBL09400
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                         TBL09410
    PUT FILE(SYSPRINT) SKIP EDIT (' STANDARD |')(A);                 TBL09420
    DO I=1 TO 15;PUT FILE(SYSPRINT) EDIT('      |') (A); END;         TBL09430
    PUT FILE(SYSPRINT) SKIP EDIT ('DEVIATION |') (A);                TBL09440
    DO I=16 TO 29; PUT FILE(SYSPRINT) EDIT (SDOSAX(2,I),' |')(F(5,1), TBL09450
        A); END;                                                     TBL09460
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                         TBL09470
    PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL09480
        '_____________________________________________________',TBL09490
        '_________________________') (A,A,A);                        TBL09500
PROSAX1 : PROCEDURE (CA);                                            TBL09520
    DCL CA FIXED BINARY (31);                                        TBL09530
    DO I=1 TO 15;                                                    TBL09540
    PUT FILE(SYSPRINT) EDIT (OSVSAX(I,CA),' |')(F(5,0),A); END;      TBL09550
END PROSAX1;                                                         TBL09560
PROSAX2 : PROCEDURE (CA);                                            TBL09570
    DCL CA FIXED BINARY (31);                                        TBL09580
    DO I=16 TO 30;                                                   TBL09590
    PUT FILE(SYSPRINT) EDIT (OSVSAX(I,CA),' |')(F(5,0),A); END;      TBL09600
END PROSAX2;                                                         TBL09610
END OSVSAXC;                                                         TBL09510
* PROCESS ('OPT=1')
LDAXCA : PROCEDURE (AXLDVSAXAA,SDLDAXAA);                            TBL09720
    DECLARE   PRLDAX1 ENTRY (FIXED BINARY (31)),                     TBL00450
              PRLDAX2 ENTRY (FIXED BINARY (31));                     TBL00460
    DCL AXLDVSAXAA (30,21) FIXED BIN(31),
        (SDLDAXAA(3,29),DWT) FLOAT BIN(16);
    DO I=1 TO 29; N = AXLDVSAXAA(I,20)*(AXLDVSAXAA(I,20) - 1);       TBL09730
    IF N=0 THEN DO; SDLDAXAA(2,I)=0; GO TO A1; END;                  TBL09740
    DWT =(AXLDVSAXAA(I,20)*SDLDAXAA(2,I)-SDLDAXAA(1,I)**2)/ N;       TBL09750
    IF DWT < 0 THEN DO; SDLDAXAA(2,I) =-1; GO TO A1; END;            TBL09760
    SDLDAXAA(2,I) = SQRT(DWT);                                       TBL09770
A1: IF AXLDVSAXAA(I,20)=0 THEN DO; SDLDAXAA(1,I)=0; GO TO A2; END;   TBL09780
    SDLDAXAA(1,I) = SDLDAXAA(1,I) / AXLDVSAXAA(I,20);                TBL09790
    AXLDVSAXAA(30,20) = AXLDVSAXAA(30,20) + AXLDVSAXAA(I,20);        TBL09800
A2: AXLDVSAXAA(30,21) = AXLDVSAXAA(30,21) + AXLDVSAXAA(I,21);        TBL09810
    END;                                                             TBL09820
```

```
PUT FILE(SYSPRINT) PAGE LINE(6) EDIT ('AXLE LOAD VERSUS AXLE PLAC',TBL09830
     'EMENT') (COLUMN(43),A,A);                                    TBL09840
PUT FILE(SYSPRINT) SKIP(2) EDIT ('TANDEM SPACING IS 40 INCHES OR ',TBL09850
     'LESS') (COLUMN(41),A,A);                                     TBL09860
PUT FILE(SYSPRINT) SKIP EDIT ('(AASHO CATEGORIES)')(COLUMN(49),A); TBL09870
PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE,BITAN',TBL09880
     'DEM AND TRITANDEM AXLES')  (COLUMN(31),A,A);                 TBL09890
PUT FILE(SYSPRINT) SKIP(5) EDIT ('   AXLE   | UNDER','AXLE PLACEM',TBL09900
     'ENT       ') (A,X(36),A,A);                                 TBL09910
PUT FILE(SYSPRINT) SKIP EDIT ('   LOAD   |2 TONS') (A);           TBL09920
PUT FILE(SYSPRINT) SKIP EDIT ('  (KIPS)  |  110    110    111    ',TBL09930
     '120   1111   1210   1120   1300  11111  12110  11210  11120',TBL09940
     '  13100  11300  12200')(A,A,A);                              TBL09950
PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL09960
     '_____________________________________________________',     TBL09970
     '_________________________')(A,A,A);                         TBL09980
PUT FILE(SYSPRINT) SKIP EDIT (' UNDER 1  |')(A); CALL PRLDAX1(1);  TBL09990
PUT FILE(SYSPRINT) SKIP EDIT ('  1 -  3  |')(A); CALL PRLDAX1(2);  TBL10000
PUT FILE(SYSPRINT) SKIP EDIT ('  3 -  5  |')(A); CALL PRLDAX1(3);  TBL10010
PUT FILE(SYSPRINT) SKIP EDIT ('  5 -  7  |')(A); CALL PRLDAX1(4);  TBL10020
PUT FILE(SYSPRINT) SKIP EDIT ('  7 -  9  |')(A); CALL PRLDAX1(5);  TBL10030
PUT FILE(SYSPRINT) SKIP EDIT ('  9 - 11  |')(A); CALL PRLDAX1(6);  TBL10040
PUT FILE(SYSPRINT) SKIP EDIT (' 11 - 13  |')(A); CALL PRLDAX1(7);  TBL10050
PUT FILE(SYSPRINT) SKIP EDIT (' 13 - 15  |')(A); CALL PRLDAX1(8);  TBL10060
PUT FILE(SYSPRINT) SKIP EDIT (' 15 - 17  |')(A); CALL PRLDAX1(9);  TBL10070
PUT FILE(SYSPRINT) SKIP EDIT (' 17 - 19  |')(A); CALL PRLDAX1(10); TBL10080
PUT FILE(SYSPRINT) SKIP EDIT (' 19 - 21  |')(A); CALL PRLDAX1(11); TBL10090
PUT FILE(SYSPRINT) SKIP EDIT (' 21 - 23  |')(A); CALL PRLDAX1(12); TBL10100
PUT FILE(SYSPRINT) SKIP EDIT (' 23 - 25  |')(A); CALL PRLDAX1(13); TBL10110
PUT FILE(SYSPRINT) SKIP EDIT (' 25 - 27  |')(A); CALL PRLDAX1(14); TBL10120
PUT FILE(SYSPRINT) SKIP EDIT (' 27 - 29  |')(A); CALL PRLDAX1(15); TBL10130
PUT FILE(SYSPRINT) SKIP EDIT (' 29 - 31  |')(A); CALL PRLDAX1(16); TBL10140
PUT FILE(SYSPRINT) SKIP EDIT (' 31 - 33  |')(A); CALL PRLDAX1(17); TBL10150
PUT FILE(SYSPRINT) SKIP EDIT (' 33 - 35  |')(A); CALL PRLDAX1(18); TBL10160
PUT FILE(SYSPRINT) SKIP EDIT (' OVER 35  |')(A); CALL PRLDAX1(19); TBL10170
PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |')(A);                   TBL10180
PUT  FILE(SYSPRINT) EDIT (('      |' DO I=1 TO 15)) (A);          TBL10190
PUT FILE(SYSPRINT) SKIP EDIT ('  AXLES   |')(A); CALL PRLDAX1(20); TBL10200
PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |')(A);                   TBL10210
PUT  FILE(SYSPRINT) EDIT (('      |' DO I=1 TO 15)) (A);          TBL10220
PUT FILE(SYSPRINT) SKIP EDIT (' VEHICLES |')(A); CALL PRLDAX1(21); TBL10230
PUT FILE(SYSPRINT) SKIP EDIT('MEAN AXLE |')(A);                   TBL10240
PUT  FILE(SYSPRINT) EDIT (('      |' DO I=1 TO 15)) (A);          TBL10250
PUT FILE(SYSPRINT) SKIP EDIT ('  WEIGHT  |',(SDLDAXAA(1,I),' |'   TBL10260
     DO I=1 TO 15)) (A,15 (F(5,1),A));                            TBL10270
PUT FILE(SYSPRINT) SKIP EDIT (' STANDARD |',('      |' DO I=1 TO  TBL10280
     15)) (A);                                                    TBL10290
PUT FILE(SYSPRINT) SKIP EDIT ('DEVIATION |',(SDLDAXAA(2,I),' |'   TBL10300
     DO I=1 TO 15)) (A,15 (F(5,1),A));                            TBL10310
PUT FILE(SYSPRINT) SKIP(0) EDIT(('____' DO I=1 TO 29)) (A);       TBL10320
PUT FILE(SYSPRINT) PAGE LINE(6) EDIT ('AXLE LOAD VERSUS AXLE PLAC',TBL10330
     'EMENT (CONTINUED FROM LAST PAGE)') (COLUMN(30),A,A);        TBL10340
PUT FILE(SYSPRINT) SKIP(2) EDIT ('TANDEM SPACING IS 40 INCHES OR ',TBL10350
     'LESS') (COLUMN(41),A,A);                                    TBL10360
PUT FILE(SYSPRINT) SKIP EDIT ('(AASHO CATEGORIES)')(COLUMN(49),A); TBL10370
PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE,BITAN',TBL10380
     'DEM AND TRITANDEM AXLES')  (COLUMN(31),A,A);                TBL10390
PUT FILE(SYSPRINT) SKIP(5) EDIT ('   AXLE   |','AXLE PLACEMENT')  TBL10400
     (A,X(42),A);                                                 TBL10410
PUT FILE(SYSPRINT) SKIP EDIT ('   LOAD   |','     ')(A,X(98),A);  TBL10420
PUT FILE(SYSPRINT) SKIP EDIT ('  (KIPS)  |111111 121110 112110 11',TBL10430
```

```
      '1210 111120 122100 112200 121200 132000 123000 131100 11310',TBL10440
      '0 111300 OVER 6  TOTALS ') (A);                              TBL10450
    PUT FILE(SYSPRINT) SKIP(0) EDIT (('____' DO I=1 TO 29)) (A);    TBL10460
    PUT FILE(SYSPRINT) SKIP EDIT (' UNDER 1  |')(A); CALL PRLDAX2(1);  TBL10470
    PUT FILE(SYSPRINT) SKIP EDIT ('  1 -  3  |')(A); CALL PRLDAX2(2);  TBL10480
    PUT FILE(SYSPRINT) SKIP EDIT ('  3 -  5  |')(A); CALL PRLDAX2(3);  TBL10490
    PUT FILE(SYSPRINT) SKIP EDIT ('  5 -  7  |')(A); CALL PRLDAX2(4);  TBL10500
    PUT FILE(SYSPRINT) SKIP EDIT ('  7 -  9  |')(A); CALL PRLDAX2(5);  TBL10510
    PUT FILE(SYSPRINT) SKIP EDIT ('  9 - 11  |')(A); CALL PRLDAX2(6);  TBL10520
    PUT FILE(SYSPRINT) SKIP EDIT (' 11 - 13  |')(A); CALL PRLDAX2(7);  TBL10530
    PUT FILE(SYSPRINT) SKIP EDIT (' 13 - 15  |')(A); CALL PRLDAX2(8);  TBL10540
    PUT FILE(SYSPRINT) SKIP EDIT (' 15 - 17  |')(A); CALL PRLDAX2(9);  TBL10550
    PUT FILE(SYSPRINT) SKIP EDIT (' 17 - 19  |')(A); CALL PRLDAX2(10); TBL10560
    PUT FILE(SYSPRINT) SKIP EDIT (' 19 - 21  |')(A); CALL PRLDAX2(11); TBL10570
    PUT FILE(SYSPRINT) SKIP EDIT (' 21 - 23  |')(A); CALL PRLDAX2(12); TBL10580
    PUT FILE(SYSPRINT) SKIP EDIT (' 23 - 25  |')(A); CALL PRLDAX2(13); TBL10590
    PUT FILE(SYSPRINT) SKIP EDIT (' 25 - 27  |')(A); CALL PRLDAX2(14); TBL10600
    PUT FILE(SYSPRINT) SKIP EDIT (' 27 - 29  |')(A); CALL PRLDAX2(15); TBL10610
    PUT FILE(SYSPRINT) SKIP EDIT (' 29 - 31  |')(A); CALL PRLDAX2(16); TBL10620
    PUT FILE(SYSPRINT) SKIP EDIT (' 31 - 33  |')(A); CALL PRLDAX2(17); TBL10630
    PUT FILE(SYSPRINT) SKIP EDIT (' 33 - 35  |')(A); CALL PRLDAX2(18); TBL10640
    PUT FILE(SYSPRINT) SKIP EDIT (' OVER 35  |')(A); CALL PRLDAX2(19); TBL10650
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |') (A);                TBL10660
    PUT FILE(SYSPRINT) EDIT (('      |' DO I=1 TO 15)) (A);         TBL10670
    PUT FILE(SYSPRINT) SKIP EDIT ('  AXLES   |')(A);CALL PRLDAX2(20); TBL10680
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |',('      |' DO I=1 TO  TBL10690
      15)) (A);                                                     TBL10700
    PUT FILE(SYSPRINT) SKIP EDIT (' VEHICLES |')(A); CALL PRLDAX2(21); TBL10710
    PUT FILE(SYSPRINT) SKIP EDIT ('MEAN AXLE |',('      |' DO I=I TO  TBL10720
      15))(A);                                                      TBL10730
    PUT  FILE(SYSPRINT) SKIP EDIT ('  WEIGHT  |',(SDLDAXAA(1,I),' |'  TBL10740
      DO I=16 TO 29)) (A,15 (F(5,1),A));                            TBL10750
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                        TBL10760
    PUT FILE(SYSPRINT) SKIP EDIT (' STANDARD |',('      |' DO I=1 TO  TBL10770
      15))(A);                                                      TBL10780
    PUT  FILE(SYSPRINT) SKIP EDIT ('DEVIATION |',(SDLDAXAA(2,I),' |'  TBL10790
      DO I=16 TO 29)) (A,15 (F(5,1),A));                            TBL10800
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                        TBL10810
    PUT FILE(SYSPRINT) SKIP(0) EDIT (('____' DO I=1 TO 29)) (A);    TBL10820
PRLDAX1 : PROCEDURE (CA);                                           TBL09620
    DECLARE CA FIXED BINARY(31);                                    TBL09630
    DO I=1 TO 15;                                                   TBL09640
    PUT FILE(SYSPRINT) EDIT (AXLDVSAXAA(I,CA),' |')(F(5,0),A); END; TBL09650
END PRLDAX1;                                                        TBL09660
PRLDAX2 : PROCEDURE (CA);                                           TBL09670
    DECLARE CA FIXED BINARY(31);                                    TBL09680
    DO I=16 TO 30;                                                  TBL09690
    PUT FILE(SYSPRINT) EDIT (AXLDVSAXAA(I,CA),' |')(F(5,0),A); END; TBL09700
END PRLDAX2;                                                        TBL09710
END LDAXCA;                                                         TBL1DB30
* PROCESS ('OPT=1')
LDAXCK : PROCEDURE (AXLDVSAX,SDLDAX);                               TBL10940
    DECLARE   PRLDAX3 ENTRY (FIXED BINARY (31)),                    TBL00470
              PRLDAX4 ENTRY (FIXED BINARY (31));                    TBL00480
    DCL AXLDVSAX(30,21) FIXED BIN(31),
      (SDLDAX(3,29),DWT) FLOAT BIN(16);
    DO I=1 TO 29; N = AXLDVSAX  (I,20)*(AXLDVSAX  (I,20) - 1);      TBL10950
    IF N=0 THEN DO; SDLDAX  (2,I)=0; GO TO A1; END;                 TBL10960
    DWT =(AXLDVSAX  (I,20)*SDLDAX  (2,I)-SDLDAX  (1,I)**2)/ N;      TBL1097D
    IF DWT < 0 THEN DO; SDLDAX  (2,I) = 0; GO TO A1; END;           TBL109B0
    SDLDAX  (2,I) = SQRT(DWT);                                      TBL10990
```

```
A1: IF AXLDVSAX  (I,20)=0 THEN DO; SDLDAX  (1,I)=0; GO TO A2; END;      TBL11000
    SDLDAX  (1,I) = SDLDAX  (1,I) / AXLDVSAX  (I,20);                   TBL11010
    AXLDVSAX  (30,20) = AXLDVSAX  (30,20) + AXLDVSAX  (I,20);           TBL11020
A2: AXLDVSAX  (30,21) = AXLDVSAX  (30,21) + AXLDVSAX  (I,21);           TBL11030
    END;                                                                TBL11040
    PUT FILE(SYSPRINT) PAGE LINE(6) EDIT ('AXLE LOAD VERSUS AXLE PLAC',TBL11050
         'EMENT') (COLUMN(43),A,A);                                     TBL11060
    PUT FILE(SYSPRINT) SKIP(2) EDIT('TANDEM SPACING IS 40 INCHES TO 1',TBL11070
         '20 INCHES') (COLUMN(38),A,A);                                 TBL11080
    PUT FILE(SYSPRINT) SKIP EDIT ('(KENTUCKY CATEGORIES)')              TBL11090
         (COLUMN (47),A);                                               TBL11100
    PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE,BITAN',TBL11110
         'DEM AND TRITANDEM AXLES')  (COLUMN(31),A,A);                  TBL11120
    PUT FILE(SYSPRINT) SKIP(5) EDIT ('   AXLE   | UNDER','AXLE PLACEM',TBL11130
         'ENT       ') (A,X(36),A,A);                                  TBL11140
    PUT FILE(SYSPRINT) SKIP EDIT ('   LOAD   |2 TONS') (A);            TBL11150
    PUT FILE(SYSPRINT) SKIP EDIT ('  (KIPS)  |  110    110    111    ',TBL11160
         '120   1111   1210   1120   I300  11111  12110  11210  11120',TBL11170
         '  13100  11300  12200')(A,A,A);                               TBL11180
    PUT FILE(SYSPRINT) SKIP(0) EDIT ('_____________________________',TBL11190
         '_____________________________________________',TBL11200
         '_________________________')(A,A,A);                           TBL11210
    PUT FILE(SYSPRINT) SKIP EDIT (' UNDER 1  |')(A); CALL PRLDAX3(1);  TBL11220
    PUT FILE(SYSPRINT) SKIP EDIT ('  1 -  3  |')(A); CALL PRLDAX3(2);  TBL11230
    PUT FILE(SYSPRINT) SKIP EDIT ('  3 -  5  |')(A); CALL PRLDAX3(3);  TBL11240
    PUT FILE(SYSPRINT) SKIP EDIT ('  5 -  7  |')(A); CALL PRLDAX3(4);  TBL11250
    PUT FILE(SYSPRINT) SKIP EDIT ('  7 -  9  |')(A); CALL PRLDAX3(5);  TBL11260
    PUT FILE(SYSPRINT) SKIP EDIT ('  9 - 11  |')(A); CALL PRLDAX3(6);  TBL11270
    PUT FILE(SYSPRINT) SKIP EDIT (' 11 - 13  |')(A); CALL PRLDAX3(7);  TBL11280
    PUT FILE(SYSPRINT) SKIP EDIT (' 13 - 15  |')(A); CALL PRLDAX3(8);  TBL11290
    PUT FILE(SYSPRINT) SKIP EDIT (' 15 - 17  |')(A); CALL PRLDAX3(9);  TBL11300
    PUT FILE(SYSPRINT) SKIP EDIT (' 17 - 19  |')(A); CALL PRLDAX3(10); TBL11310
    PUT FILE(SYSPRINT) SKIP EDIT (' 19 - 21  |')(A); CALL PRLDAX3(11); TBL11320
    PUT FILE(SYSPRINT) SKIP EDIT (' 21 - 23  |')(A); CALL PRLDAX3(12); TBL11330
    PUT FILE(SYSPRINT) SKIP EDIT (' 23 - 25  |')(A); CALL PRLDAX3(13); TBL11340
    PUT FILE(SYSPRINT) SKIP EDIT (' 25 - 27  |')(A); CALL PRLDAX3(14); TBL11350
    PUT FILE(SYSPRINT) SKIP EDIT (' 27 - 29  |')(A); CALL PRLDAX3(15); TBL11360
    PUT FILE(SYSPRINT) SKIP EDIT (' 29 - 31  |')(A); CALL PRLDAX3(16); TBL11370
    PUT FILE(SYSPRINT) SKIP EDIT (' 31 - 33  |')(A); CALL PRLDAX3(17); TBL11380
    PUT FILE(SYSPRINT) SKIP EDIT (' 33 - 35  |')(A); CALL PRLDAX3(18); TBL11390
    PUT FILE(SYSPRINT) SKIP EDIT (' OVER 35  |')(A); CALL PRLDAX3(19); TBL11400
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |')(A);                   TBL11410
    PUT  FILE(SYSPRINT) EDIT (('       |' DO I=1 TO 15)) (A);         TBL11420
    PUT FILE(SYSPRINT) SKIP EDIT ('  AXLES   |')(A); CALL PRLDAX3(20); TBL11430
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |')(A);                   TBL11440
    PUT  FILE(SYSPRINT) EDIT (('       |' DO I=1 TO 15)) (A);         TBL11450
    PUT FILE(SYSPRINT) SKIP EDIT (' VEHICLES |')(A); CALL PRLDAX3(21); TBL11460
    PUT FILE(SYSPRINT) SKIP EDIT('MEAN AXLE |')(A);                     TBL11470
    PUT  FILE(SYSPRINT) EDIT (('       |' DO I=1 TO 15)) (A);         TBL11480
    PUT FILE(SYSPRINT) SKIP EDIT ('  WEIGHT  |',(SDLDAX  (1,I),' |'    TBL11490
         DO I=1 TO 15)) (A,15 (F(5,1),A));                              TBL11500
    PUT FILE(SYSPRINT) SKIP EDIT (' STANDARD |',('       |' DO I=1 TO  TBL11510
         15)) (A);                                                      TBL11520
    PUT FILE(SYSPRINT) SKIP EDIT ('DEVIATION |',(SDLDAX  (2,I),' |'    TBL11530
         DO I=1 TO 15)) (A,15 (F(5,1),A));                              TBL11540
    PUT FILE(SYSPRINT) SKIP(0) EDIT(('____' DO I=1 TO 29)) (A);         TBL11550
    PUT FILE(SYSPRINT) PAGE LINE(6) EDIT ('AXLE LOAD VERSUS AXLE PLAC',TBL11560
         'EMENT (CONTINUED FROM LAST PAGE)') (COLUMN(30),A,A);          TBL11570
    PUT FILE(SYSPRINT) SKIP(2) EDIT('TANDEM SPACING IS 40 INCHES TO 1',TBL11580
         '20 INCHES') (COLUMN(38),A,A);                                 TBL11590
    PUT FILE(SYSPRINT) SKIP EDIT ('(KENTUCKY CATEGORIES)')              TBL11600
```

```
      (COLUMN (47),A);                                                    TBL11610
    PUT FILE(SYSPRINT) SKIP(2) EDIT ('1,2 AND 3 INDICATE SINGLE,BITAN',TBL11620
      'DEM AND TRITANDEM AXLES') (COLUMN(31),A,A);                        TBL11630
    PUT FILE(SYSPRINT) SKIP(5) EDIT ('   AXLE   |','AXLE PLACEMENT')     TBL11640
      (A,X(42),A);                                                        TBL11650
    PUT FILE(SYSPRINT) SKIP EDIT ('   LOAD   |','     ')(A,X(98),A);     TBL11660
    PUT FILE(SYSPRINT) SKIP EDIT ('  (KIPS)  |111111 121110 112110 11',TBL11670
      '1210 111120 122100 112200 121200 132000 123000 131100 11310',TBL11680
      '0 111300 OVER 6  TOTALS ') (A);                                    TBL11690
    PUT FILE(SYSPRINT) SKIP(0) EDIT (('____' DO I=1 TO 29)) (A);          TBL11700
    PUT FILE(SYSPRINT) SKIP EDIT (' UNDER 1  |')(A); CALL PRLDAX4(1);     TBL11710
    PUT FILE(SYSPRINT) SKIP EDIT ('  1 -  3  |')(A); CALL PRLDAX4(2);     TBL11720
    PUT FILE(SYSPRINT) SKIP EDIT ('  3 -  5  |')(A); CALL PRLDAX4(3);     TBL11730
    PUT FILE(SYSPRINT) SKIP EDIT ('  5 -  7  |')(A); CALL PRLDAX4(4);     TBL11740
    PUT FILE(SYSPRINT) SKIP EDIT ('  7 -  9  |')(A); CALL PRLDAX4(5);     TBL11750
    PUT FILE(SYSPRINT) SKIP EDIT ('  9 - 11  |')(A); CALL PRLDAX4(6);     TBL11760
    PUT FILE(SYSPRINT) SKIP EDIT (' 11 - 13. |')(A); CALL PRLDAX4(7);     TBL11770
    PUT FILE(SYSPRINT) SKIP EDIT (' 13 - 15  |')(A); CALL PRLDAX4(8);     TBL11780
    PUT FILE(SYSPRINT) SKIP EDIT (' 15 - 17  |')(A); CALL PRLDAX4(9);     TBL11790
    PUT FILE(SYSPRINT) SKIP EDIT (' 17 - 19  |')(A); CALL PRLDAX4(10);    TBL11800
    PUT FILE(SYSPRINT) SKIP EDIT (' 19 - 21  |')(A); CALL PRLDAX4(11);    TBL11810
    PUT FILE(SYSPRINT) SKIP EDIT (' 21 - 23  |')(A); CALL PRLDAX4(12);    TBL11820
    PUT FILE(SYSPRINT) SKIP EDIT (' 23 - 25  |')(A); CALL PRLDAX4(13);    TBL11830
    PUT FILE(SYSPRINT) SKIP EDIT (' 25 - 27  |')(A); CALL PRLDAX4(14);    TBL11840
    PUT FILE(SYSPRINT) SKIP EDIT (' 27 - 29  |')(A); CALL PRLDAX4(15);    TBL11850
    PUT FILE(SYSPRINT) SKIP EDIT (' 29 - 31  |')(A); CALL PRLDAX4(16);    TBL11860
    PUT FILE(SYSPRINT) SKIP EDIT (' 31 - 33  |')(A); CALL PRLDAX4(17);    TBL11870
    PUT FILE(SYSPRINT) SKIP EDIT (' 33 - 35  |')(A); CALL PRLDAX4(18);    TBL11880
    PUT FILE(SYSPRINT) SKIP EDIT (' OVER 35  |')(A); CALL PRLDAX4(19);    TBL11890
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |') (A);                     TBL11900
    PUT FILE(SYSPRINT) EDIT (('      |' DO I=1 TO 15)) (A);              TBL11910
    PUT FILE(SYSPRINT) SKIP EDIT ('  AXLES   |')(A);CALL PRLDAX4(20);     TBL11920
    PUT FILE(SYSPRINT) SKIP EDIT ('  TOTAL   |',('      |' DO I=1 TO    TBL11930
      15)) (A);                                                           TBL11940
    PUT FILE(SYSPRINT) SKIP EDIT (' VEHICLES |')(A); CALL PRLDAX4(21);    TBL11950
    PUT FILE(SYSPRINT) SKIP EDIT ('MEAN AXLE |',('      |' DO I=1 TO    TBL11960
      15))(A);                                                            TBL11970
    PUT  FILE(SYSPRINT) SKIP EDIT ('  WEIGHT  |',(SDLDAX  (1,I),' |'     TBL11980
      DO I=16 TO 29)) (A,15 (F(5,1),A));                                  TBL11990
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                              TBL12000
    PUT FILE(SYSPRINT) SKIP EDIT (' STANDARD |',('      |' DO I=1 TO    TBL12010
      15))(A);                                                            TBL12020
    PUT  FILE(SYSPRINT) SKIP EDIT ('DEVIATION |',(SDLDAX  (2,I),' |'     TBL12030
      DO I=16 TO 29)) (A,15 (F(5,1),A));                                  TBL12040
    PUT FILE(SYSPRINT) EDIT (' ---- |') (A);                              TBL12050
    PUT FILE(SYSPRINT) SKIP(0) EDIT (('____' DO I=1 TO 29)) (A);          TBL12060
PRLDAX3 : PROCEDURE (CA);                                                 TBL10840
    DECLARE CA FIXED BINARY(31);                                          TBL10850
    DO I=1 TO 15;                                                         TBL10860
    PUT FILE(SYSPRINT) EDIT (AXLDVSAX  (I,CA),' |')(F(5,0),A); END;       TBL10870
END PRLDAX3;                                                              TBL10880
PRLDAX4 : PROCEDURE (CA);                                                 TBL10890
    DECLARE CA FIXED BINARY(31);                                          TBL10900
    DO I=16 TO 30;                                                        TBL10910
    PUT FILE(SYSPRINT) EDIT (AXLDVSAX  (I,CA),' |')(F(5,0),A); END;       TBL10920
END PRLDAX4;                                                              TBL10930
END LDAXCK;                                                               TBL12070
/*
//GO.SYSUDUMP  DD  SYSOUT=A
//GO.TAPE  DD  UNIT=2400,DISP=(OLD,KEEP),DCB=(BLKSIZE=2500,LRECL=25,
//       DEN=2,RECFM=FB),VOL=SER=HIWY09,
```

```
//         LABEL=(1,SL),DSNAME=TAPE01
/*
```