Interim Research Report
KTC-94-6

BARGE IMPACT LOADS FOR THE MAYSVILLE BRIDGE

by

Michael W. Whitney
Regional Structural Engineer, Federal Highway Administration
(On educational leave to pursue a Ph.D degree in Civil Engineering)

Issam E. Harik
Professor of Civil Engineering

James J. Griffin
Research Assistant

and

David L. Allen
Chief Research Engineer

Kentucky Transportation Center
College of Engineering, University of Kentucky

in cooperation with

Transportation Cabinet
Commonwealth of Kentucky

and

Federal Highway Administration
U.S. Department of Transportation

February 1994

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
KTC-94-6		

4. Title and Subtitle

BARGE IMPACT LOADS FOR THE MAYSVILLE BRIDGE

5. Report Date
February 1994

6. Performing Organization Code

7. Author(s)
M.W. Whitney, I.E. Harik, J.J. Griffin, D.L. Allen

8. Performing Organization Report No.
KTC-94-6

9. Performing Organization Name and Address

Kentucky Transportation Center
College of Engineering
University of Kentucky
Lexington, KY 40506-0043

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
FRT-65

12. Sponsoring Agency Name and Address
Kentucky Transportation Cabinet
State Office Building
Frankfort, KY 40622

13. Type of Report and Period Covered
Interim

14. Sponsoring Agency Code

15. Supplementary Notes
Prepared in cooperation with the Kentucky Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration

16. Abstract

This report provides the barge equivalent static load distribution data needed to apply method II of vessel impact design for the Maysville, Kentucky bridge over the Ohio River. The information provided in this report is in accordance with the *AASHTO Guide Specification and Commentary for Vessel Collision Design of Highway Bridges*.

A computer program was written to process the database and calculate the probability based length, width, and capacity for each barge category. Additionally, a second computer program was written to calculate the probability based number of barges in a flotilla column and row, and subsequently categorize that flotilla based upon the barge length and width categories designated by the U.S. Army Corps of Engineers. The equivalent static impact loads were then calculated using the probability based flotilla sizes and tonnages.

The results indicated 12 barge categories occurring along the Maysville section of the Ohio River. The associated frequencies and impact loads are reported. The equivalent static load for the usually neglected 290 x 54 foot barge was calculated to be a maximum of 8,140 kips with a significant frequency of occurrence of 205 downbound passages per year.

17. Key Words		18. Distribution Statement
Downbound	Upbound	Unlimited with approval of Kentucky Transportation Cabinet
Flotilla	Barge	
Probability	Impact Load	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	64	

Table of Contents

Introduction

This report provides the barge equivalent static load distribution data needed to apply method II of vessel impact design (reference 1) for the Maysville, Kentucky bridge over the Ohio River. The information provided in this report is in accordance with the *AASHTO Guide Specification and Commentary for Vessel Collision Design of Highway Bridges* (reference 1). The results generated in this report are based on statistical data obtained from the U.S. Coast Guard, the U.S. Army Corps of Engineers, and the American Waterways Operators.

The AASHTO Guide Specification recommends that the impact loads from transiting flotillas be applied at the 2% flow as the water elevation. For the Maysville bridge only, the tower piers are located in the waterway at the this elevation. Therefore, only the tower piers need resist the flotilla impact loads.

Probability Based Impact Loads for the Tower Piers

In order to calculate the equivalent static impact loads necessary to apply method II for vessel impact design, the barge flotillas currently using the Maysville section of the Ohio River needed to be classified into discrete categories. However, the Army Corps of Engineers reports that there are approximately 2,000 barge sizes and types currently using the U.S. inland waterway system. In addition, barge flotillas can be comprised of almost any combination and number of these barge sizes and types.

Typically though, the flotilla sizes are limited by the dimensions of the locks on the waterways that the flotilla will pass through and by practical considerations such as lack of maneuverability for excessively large flotillas, etc. Also, flotillas are generally madeup of mostly the same barge size and type. Nevertheless, there still is a very large variation in the flotillas using the Maysville section of the Ohio River; therefore, a probability approach was used to calculate the size, tonnage, and number of barges makingup 12 flotilla categories.

It should be noted that there are 24 possible flotilla categories; however, only 12 occur on the Maysville section of the Ohio River. The 24 flotilla categories were based upon the U.S. Army Corps of Engineers designation as presented in Tables 1 and 2. Flotilla category size ranges currently using the Maysville section of the Ohio River (Mile 411) are given in Table 3 (columns 1 and 2, respectively).

The impact loads and their associated frequencies are also given in Table 3 for

the west and east tower piers (columns 4-6). The methods and databases used to calculate the values given in the Table 3 are given in the following sections of this report.

Probability Based Barge Sizes and Tonnages

The barge tonnages and sizes for the 12 categories are based on the information contained in the Waterborne Transportation Lines of the United States database. The database contains sizes and tonnages of every barge registered to operate in the U.S. A computer program (given in Appendix I) was written to process the database and calculate the tonnages and sizes to be assigned to the barges comprising a flotilla category. The computer calculations were based on the following assumptions:

1. The variation of the barge sizes and tonnages within a category can be represented by a normal distribution.

2. The barges using the Kentucky waterways do not exceed a loaded draft of 15.2-ft.

   The draft cutoff of 15.2-ft was based on information from the U.S. Coast Guard that barges with a draft in excess of 12-ft do not typically operate on Kentucky waterways. The 15.2-ft value was used to include some barges in the database that could conceivably operate on the Kentucky waterways during high water conditions. This will lead to reasonably conservative results.

3. The minimum of the following values is used:

   - maximum sizes, and tonnages encountered for a category within the database.
   - average sizes and tonnages plus two standard deviations calculated for a category.

   Since the variation of the barge sizes and tonnages within a category could be represented by a normal distribution, use of the average plus two standard deviations assures that the barge sizes and tonnages assigned to a category have only a 4.5% chance of being exceeded. In the cases where the maximum values within a category are used, there is 0% chance that the sizes and tonnages will be exceeded since the database contains all barges operating within the U.S.

4. Only barges typically operating on the Mississippi River System and the Gulf Coast Intercostal Waterway will be used in the calculations.

5. The barge selfweight could be linearly interpolated from the relationship:

$$self\ weight\ = (cargo\ capacity\ ) * \left[\frac{light\ draft}{loaded\ draft\ -\ light\ draft} \right] \qquad (1)$$

The results from the computer program calculations are given in Appendix II.

Probability Based Flotilla Column and Row Count

The number of barges in a flotilla for the 12 categories is based on the information contained in the 1992 Performance Monitoring System database. The database contains information on the total number of barges by category and the total number of flotillas and their dimensions using the locks of the U.S. Waterways in 1992. It should be noted that the flotillas are not necessarily comprised of only one barge size or type. In addition, the database only contains information on the total number of barges in each flotilla and the dimensions of each flotilla. Therefore, certain assumptions must be made in order to determine the number of barges in each row and column of the flotillas and the size of the barges makingup the flotilla.

A computer program (given in Appendix III) was written to process the database and calculate the number of barges to be assigned to the rows and columns of the 12 flotilla categories. The computer calculations were based on the following assumptions:

1. The variation of the number of barges comprising the rows and columns of a flotilla within a category can be represented by a normal distribution.

2. The flotilla width varies in regular increments and therefore the number of barges in the rows is determined first.

3. Barge widths do not *typically* exceed 55 feet.

4. The minimum of the following values is used:

   - maximum number of barges makingup the rows and columns encountered for a category within the database.
   - average number of barges makingup the rows and columns plus two standard deviations calculated for a category.

Since it was assumed that the variation of the number of barges

comprising the flotilla category rows and columns could be represented by a normal distribution, using the average plus two standard deviations indicates there is only a 4.5% chance of the values used being exceeded on a yearly basis.

5. Non-integer values for the number of barges comprising a flotilla column or row are acceptable since method II is a probability based analysis procedure.

6. Flotilla column lengths include the possibility of a barge attached to the side of the tow boat. Since tow boat tonnages are generally lower than barge tonnages, it is more conservative to replace the tow boat with a barge.

The flotilla frequency distribution for the Maysville section of the Ohio River was determined by dividing the total number of barges for each category by the average number of barges comprising each of the flotilla categories. The average number was used in place of the average plus two standard deviations since it would result in a more conservative flotilla frequency distribution. The total numbers of barges by category are given in Appendix IV. Some barge types do not occur as flotillas, rather they are incorporated in flotillas which are comprised primarily of other barge types. These categories are assigned a "zero" flotilla frequency.

For the Maysville section, upbound barges operate at only 31% of cargo capacity and travel at maximum absolute velocities (barge minus river velocity) of approximately four knots (7 fps, 5 mph). On the other hand, downbound barges travel at 93% of cargo capacities with absolute velocities (barge plus river velocity) of ten knots (17 fps, 12 mph). Consequently, impact loads and barge counts neglect upbound barge traffic since impact loads from upbound barges are insignificant compared to downbound barges.

Location of Tower Pier Impact Loads

For the substructure stability design, Section 3.15.1 of the *AASHTO Guide Specification and Commentary* (reference 1) is unclear as to which river flow condition the impact load should be applied. However, it seems the intent of Section 3.15.1 is to require the 2% flow elevation. The 2% flow elevation, determined from daily river flow data, is the elevation the river exceeds just 2% of the time. The 2% flow elevation should not be confused with a 2% flood event elevation, which is the elevation the river has a 2% probability of reaching for any given year.

It is recommended that the concentrated impact load be applied to the tower

piers at the 2% flow elevation of 496.5-ft. In addition, reference 1 allows for the local or impacted pier to be designed with the barge impact load applied as a uniformly distributed load. The recommended starting elevation and length of the uniform barge impact loads are given in Table 3 (columns 7 and 8) by flotilla category. These elevations are based on barge size data from the Waterborne Transportation Lines of the U.S. database (WTLUS) and the information provided in the *AASHTO Guide Specification and Commentary for Vessel Collision Design of Highway Bridges*. The elevations assume that the barge contacts only the tower pier columns and does not contact the substructure (e.g., pile footing, etc.).

Impact Force Equations

Currently, reference 1 provides a simple method for calculating the equivalent static barge impact force on a bridge element. The formulas are based on impact tests conducted on individual European barges. This is of concern since the tests were conducted on single barges at low velocities and not on multi-barge flotillas traveling at high velocities as found on the Ohio River.

The following exemplifies the reason for the concern. The lead barge crushing depth is required when determining the impact force. The calculations for the Ohio River barges gave crushing depths up to 39.31-ft. However, inspections of past collisions on inland waterways have shown that crushing depths rarely exceed 10-ft. These inspections have shown that energy loss occurs between individual barges due to crushing and friction. As barges crash into and ride up on each other, the amount of crushing in the lead barge is reduced, and in turn the resulting impact load is reduced.

However, in lieu of physical flotilla impact test results to provide a basis for modifying the AASHTO impact formulas (reference 1), these formulas will be used in their current form. In addition, when preliminary results from a FHWA study were used to calculate the impact load for the largest flotilla type, there was only a 15% reduction in the impact load when compared to the AASHTO impact load. Therefore, the loads are conservative but apparently not overly conservative. The calculations using the AASHTO formulas are included in Appendix V.

Minimum Impact Loads for Tower Piers

As a minimum, the AASHTO Guide Specifications require that all waterway

piers, with available water depth equal to the empty draft of a free floating barge, be designed to resist the impact of the empty barge floating with the yearly mean current velocity and elevation at the bridge location. However, the Kentucky Transportation Cabinet has established the more conservative requirement of a single barge, fully loaded or loaded to a draft equal to the available water depth, drifting at the 100-year current as the design minimum.

The design minimum barge selected was a 53-ft x 290-ft barge since it is one of the largest barges currently in use on the Ohio River, and barge traffic data indicate 205 downbound passages per year of flotillas with this barge type. The typical dimensions for the 53-ft x 290-ft barge along with other barge sizes are given in Table 4. The uniform impact load, length, and bridge pier starting elevations for the single, fully loaded barge are given in Table 5.

River Velocity

River velocity values used in the barge flotilla impact force calculations are for 2% flow at the east and west tower piers. The single free drifting barge impact forces were calculated using the 100-year flood velocity at the tower piers. River velocities were calculated by Palmer Engineering using a WSPRO analysis. However, the one-dimensional WSPRO analysis will not give the river flow directions at the tower piers necessary to determine the longitudinal and transverse components, with respect to the bridge pier, of the barge impact force. A 2-dimensional analysis, such as the University of Kentucky's FESWMS computer program, is required in order to calculate flow directions.

Vessel Velocity

The vessel transit velocity (does not include river flow velocity) used in the impact force calculations is based on data provided by the U.S. Coast Guard. The data indicated that typical vessel transit velocities were between 5 mph (7 fps, 4 knots) to 7 mph (10 fps, 6 knots). The higher value of 7 mph was used in the calculations.

Typically the total vessel velocity at the bridge pier is calculated by adding the transit velocity to the centerline river velocity and applying section 3.7 of the *AASHTO Guide Specifications* (reference 1) to reduce the centerline velocity to the value expected at the bridge piers. However, for the Maysville bridge the vessel transit path

width is equal to the navigation channel width.

The vessel velocities at the bridge piers are equal to the vessel velocities at the transit path centerline when calculated in accordance with section 3.7 of the *AASHTO Guide Specifications*. This seems conservative since the river velocity would decrease as the river bank is approached due to frictional effects. Therefore the river velocities discussed above were added to the vessel transit velocity to generate the vessel velocity at the two tower piers.

Probability of Aberrancy

Vessel accident statistics have been maintained for the last 11 years for the Ohio River. Over the past 11 years, the average probability of aberrancy for barge traffic on the Maysville bridge section of the river was 8.57×10^{-4}. It should be noted that for the years of 1990 and 1991 the average probability of aberrancy was much higher at 23.10×10^{-4}. However, *AASHTO* recommends using the long-term data. In addition, barge traffic increased only 1% from 1989 to 1990 and yet the probability of aberrancy increased three times. Consequently, the change in aberrancy rate was not related to a dramatic increase in barge traffic, which would warrant the use of short-term aberrancy rate, but rather to some short-term environmental factors (e.g., weather, low river levels, etc). Therefore, the 11 year average value of 8.57×10^{-4} should be used.

Design Barge Acceptance Criteria

For the Maysville bridge, which has a critical bridge importance classification, the acceptable annual frequency (AF_c) of collapse shall be less than or equal to 0.01 in 100 years or $AF_c = 0.0001$. The annual frequency of bridge collapse is distributed, either equally or at the designers discretion, over all piers that are located within the waterway. However, it is recommended that the AF_c be distributed to each pier based on its percentage value of the replacement cost of the structure.

For the Maysville bridge, however, only the two tower piers will be in the waterway for the 2% flow elevation. Therefore, the acceptable annual frequency of collapse for each tower pier (AF_p) should be;

$$AF_p = (1.0/2)(AF_c) = (1.0/2)(0.0001) = 0.00005$$

The summation of the annual frequencies of collapse for all barge size categories, with respect to the individual tower piers, should then be less than or equal to *0.00005*. In addition to the probability of aberrancy provided in the previous section, the data required for generating the annual frequencies of collapse for all barge size categories is provided in Table 3.

Scour Requirements

The current AASHTO Guide Specifications do not provide guidance on the application of scour to the barge impact design of bridges. However, in a letter dated September 4, 1992 the FHWA Region 4 office directed the application of the following scour conditions to the impact design using the AASHTO method II:

1. For impact loads applied at normal vessel operating conditions, two scour conditions should be evaluated. The first is the scour having a probability of 1.0, most likely only the long-term scour plus the contraction and local scour caused by a Q_5 event. The second is the maximum anticipated scour (or other critical value determined by the designer), and the probability of this scour occurring during the life of the bridge should be included in the calculations.

2. For the case of the free-floating empty barge on the 100-year flood, the maximum anticipated scour should be used.

Therefore, it is recommended that the impact loads for the loaded barge flotillas given in Table 3 be applied in conjunction with 100% of long-term scour plus the local scour caused by a Q_5 (five year return period) flood event. The impact loads for the single free-floating barge given in Table 5 should be applied with the scour caused by the Q_{100} flood event.

Conclusions

The equivalent static loads and their associated frequencies have been derived for the Maysville Bridge over the Ohio River in accordance with the requirements of method II of the *AASHTO Guide Specifications* (reference 1). The impact loads calculated using the *AASHTO* formulas are probably conservative. However, in lieu

of physical flotilla impact test results to provide a basis for modification, the *AASHTO* formulas were used in their current form.

There is a tremendous variation in the size and types of barges and flotillas in use on the Ohio River. Based on the procedures used in this report, there are currently 12 flotilla categories on the Maysville section of the Ohio River. The flotilla sizes and tonnages used to calculate the equivalent static loads for each category have at most a 4.5% chance that a flotilla will pass the Maysville Bridge with greater size or load. Finally, calculations for the equivalent static loads indicated that some categories may be combined since they result in nearly identical impact loads.

Reference

1. "Guide Specification and Commentary for Vessel Collision Design of Highway Bridges, Volume I: Final Report," *American Association of State Highway and Transportation Officials (AASHTO)*, Washington, D.C., 1991.

Table 1: Length of Barge Designation

Length of Barge	
A	less than 100 feet
B	100 to 174 feet
C	175 to 194 feet
D	195 to 199 feet
E	200 to 259 feet
F	260 to 289 feet
G	290 to 300 feet
H	greater than 300 feet

Table 2: Width of Barge Designation

Width of Barge	
A	less than 26 feet
B	26 to 34 feet
C	35 to 54 feet
D	greater than 54 feet

Table 3: Equivalent static barge impact loads and frequencies for the west and east tower piers for the Maysville, Kentucky bridge.

Flotilla Category[a] (1)	Barge Size Range for Flotilla Category (2)	Number of Barges in Flotilla Column[b] (3)	Flotilla Frequency (number of passages per year)[c] (4)	Equivalent Static Impact Force for West Tower Pier (kips) (5)	Equivalent Static Impact Force for East Tower Pier (kips) (6)	Starting Elevation of Uniform Barge Impact Load[d] (ft) (7)	Length of Uniform Barge Impact Load (ft) (8)
1 (BB)	(100-174') x (26'-34')	3.33	4	3,380	3,450	499.5	3.0
2 (BC)	(100'-174') x (35'-54')	3.42	105	7,170	7,320	499.5	3.0
3 (CC)	(175'-194') x (35'-54')	3.35	46	6,940	7,080	499.5	3.0
4 (DB)	(195'-199') x (26'-34')	5.00	25	4,230	4,330	499.5	3.0
5 (DC)	(195'-199') x (35'-54')	4.58	2076	5,640	5,760	499.5	3.0
6 (DD)	(195'-199') x (>54')	6.00	1	6,480	6,620	500.5	4.0
7 (EB)	(200'-259') x (26'-34')	5.00	1	3,610	3,700	499.5	3.0
8 (EC)	(200'-259') x (35'-54')	4.58	195	6,160	6,290	499.5	3.0
9 (FC)	(260'-289') x (35'-54')	3.35	5	7,700	7,860	499.5	3.0
10 (GC)	(290'-300') x (35'-54')	3.39	205	7,970	8,140	500.5	4.0
11 (HC)	(>300') x (35'-54')	2.00	5	8,120	8,290	500.5	4.0
12 (HD)	(>300') x (>54')	1.67	19	6,100	6,120	500.5	4.0

a: The first letter in parentheses refers to the length of barge designation as presented in Table 2, and the second letter in parentheses refers to the width of barge designation as presented in Table 3.

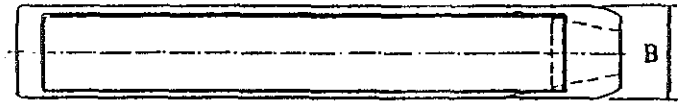
b: Non-integer values for the number of barges comprising a flotilla column are acceptable since method II is a probability based method of analysis.

c: Downbound traffic for the year 1992. Average traffic growth rate for 1991 - 1992 is -2%.

d: For both the west and east tower piers

Table 4: Typical barge size dimensions.

Length L_B (ft)	Width B_M (ft)	Depth D_V (ft)	Empty Draft D_E (ft)	Loaded Draft D_L (ft)	Depth of Bow D_B (ft)	Bow Rake Length R_L (ft)	Head Log Height H_L (ft)	Cargo Weight C_c (tons)	Empty Weight W_E (tons)	Total Weight W_L (tons)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
195	35	12	1.7	8.7	13	20	2-3	1700	200	1900
290	53	12	1.7	8.7	13	25	2-3	3700	600	4300
250	72	17	2.5	12.5	18	30	3-5	5000	1300	6300



PLAN

ELEVATION

Table 5: Equivalent static impact loads for the west and east tower piers for a single free floating 53-ft x 290-ft barge.

Uniform Barge Impact Load Starting Elevation (ft) (1)	Uniform Barge Impact Load Length (ft) (2)	Equivalent Static Impact Force West Pier (kips) (3)	Equivalent Static Impact Force East Pier (kips) (4)
500.5	4.0	2,790	2,850

Appendix I:

Computer Program for Barge Capacities

```
C*********************************************************************
C
C     READS BARGE CHARACTERISTIC FILE AND SUMS AVERAGE BARGE CAPACITIES
C     BASED ON LENGTH/WIDTH CATEGORY
C
C     CALCULATES CAPACITIES, LENGTHS, AND WIDTHS AS AVERAGE PLUS TWO
C     STANDARD DEVIATIONS
C
C     FINDS MAXIMUM CATEGORY CHARACTERISTICS
C
C     PRINTS MINIMUM OF AVERAGE BARGE CHARACTERISTIC PLUS TWO
C     STANDARD DEVIATIONS OR MAXIMUM VALUE FOUND IN CATEGORY
C
C
C         BY:MICHAEL W. WHITNEY
C            UNIVERSITY OF KENTUCKY
C            LEXINGTON, KY
C
C*********************************************************************
C
C     BLENGTH = BARGE LENGTH
C     BWIDTH = BARGE WIDTH
C     CAPACITY = BARGE CARGO CAPACITY + SELF WEIGHT
C     DRAFT = LOADED DRAFT OF THE BARGE WHICH IS ASSUMED TO BE < 15.2'
C     IAREA = LOCATION THE BARGE OPERATES (FOR KENTUCKY = 4)
C
C*********************************************************************

      IMPLICIT REAL(A-H,L-Z), INTEGER(I-K)



      OPEN (1, FILE = 'C:\F32\WATER.DAT')
      OPEN (2, FILE = 'C:\F32\WATERBRN.OUT')

C
C     CALCULATE BARGE AVERAGE AND MAXIMUM CHARACTERISTICS
C

      DO WHILE(.NOT.EOF(1))


      READ (1,*) BLENGTH,BWIDTH,CAPACITY,LOADDRAFT,LIGHTDRAFT,IAREA


1     IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GT.10..AND.
     1    BWIDTH.LT.26. .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.10.0 .AND. LOADDRAFT .GT. LIGHTDRAFT ) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          AACOUNT=AACOUNT +1
          AACAPACITY=AACAPACITY+CAPACITY
          AALENGTH=AALENGTH+BLENGTH
          AAWIDTH=AAWIDTH+BWIDTH
          AAMAXCAPACITY=MAX(AAMAXCAPACITY,CAPACITY)
          AAMAXLENGTH=MAX(AAMAXLENGTH,BLENGTH)
          AAMAXWIDTH=MAX(AAMAXWIDTH,BWIDTH)

      ENDIF

2     IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 4000.0 .AND. LOADDRAFT
     3    .GT. LIGHTDRAFT) THEN
```

```
CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          ABCOUNT=ABCOUNT +1
          ABCAPACITY=ABCAPACITY+CAPACITY
          ABLENGTH=ABLENGTH+BLENGTH
          ABWIDTH=ABWIDTH+BWIDTH
          ABMAXCAPACITY=MAX(ABMAXCAPACITY,CAPACITY)
          ABMAXLENGTH=MAX(ABMAXLENGTH,BLENGTH)
          ABMAXWIDTH=MAX(ABMAXWIDTH,BWIDTH)


      ENDIF

3     IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 5000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          ACCOUNT=ACCOUNT +1
          ACCAPACITY=ACCAPACITY+CAPACITY
          ACLENGTH=ACLENGTH+BLENGTH
          ACWIDTH=ACWIDTH+BWIDTH
          ACMAXCAPACITY=MAX(ACMAXCAPACITY,CAPACITY)
          ACMAXLENGTH=MAX(ACMAXLENGTH,BLENGTH)
          ACMAXWIDTH=MAX(ACMAXWIDTH,BWIDTH)

      ENDIF

4     IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT ) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          ADCOUNT=ADCOUNT +1
          ADLENGTH=ADLENGTH+BLENGTH
          ADWIDTH=ADWIDTH+BWIDTH
          ADCAPACITY=ADCAPACITY+CAPACITY
          ADMAXCAPACITY=MAX(ADMAXCAPACITY,CAPACITY)
          ADMAXLENGTH=MAX(ADMAXLENGTH,BLENGTH)
          ADMAXWIDTH=MAX(ADMAXWIDTH,BWIDTH)

      ENDIF

5     IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GT.10..AND.
     1    BWIDTH.LT.26. .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          BACOUNT=BACOUNT +1
          BALENGTH=BALENGTH+BLENGTH
          BAWIDTH=BAWIDTH+BWIDTH
          BACAPACITY=BACAPACITY+CAPACITY
          BAMAXCAPACITY=MAX(BAMAXCAPACITY,CAPACITY)
          BAMAXLENGTH=MAX(BAMAXLENGTH,BLENGTH)
          BAMAXWIDTH=MAX(BAMAXWIDTH,BWIDTH)

      ENDIF

6     IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN
```

```
CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          BBCOUNT=BBCOUNT +1
          BBLENGTH=BBLENGTH+BLENGTH
          BBWIDTH=BBWIDTH+BWIDTH
          BBCAPACITY=BBCAPACITY+CAPACITY
          BBMAXCAPACITY=MAX(BBMAXCAPACITY,CAPACITY)
          BBMAXLENGTH=MAX(BBMAXLENGTH,BLENGTH)
          BBMAXWIDTH=MAX(BBMAXWIDTH,BWIDTH)

      ENDIF

7     IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          BCCOUNT=BCCOUNT +1
          BCLENGTH=BCLENGTH+BLENGTH
          BCWIDTH=BCWIDTH+BWIDTH
          BCCAPACITY=BCCAPACITY+CAPACITY
          BCMAXCAPACITY=MAX(BCMAXCAPACITY,CAPACITY)
          BCMAXLENGTH=MAX(BCMAXLENGTH,BLENGTH)
          BCMAXWIDTH=MAX(BCMAXWIDTH,BWIDTH)

      ENDIF

8     IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          BDCOUNT=BDCOUNT +1
          BDLENGTH=BDLENGTH+BLENGTH
          BDWIDTH=BDWIDTH+BWIDTH
          BDCAPACITY=BDCAPACITY+CAPACITY
          BDMAXCAPACITY=MAX(BDMAXCAPACITY,CAPACITY)
          BDMAXLENGTH=MAX(BDMAXLENGTH,BLENGTH)
          BDMAXWIDTH=MAX(BDMAXWIDTH,BWIDTH)

      ENDIF

9     IF (BLENGTH.GE.175..AND.BLENGTH.LE.194..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
          CBCOUNT=CBCOUNT +1
          CBLENGTH=CBLENGTH+BLENGTH
          CBWIDTH=CBWIDTH+BWIDTH
          CBCAPACITY=CBCAPACITY+CAPACITY
          CBMAXCAPACITY=MAX(CBMAXCAPACITY,CAPACITY)
          CBMAXLENGTH=MAX(CBMAXLENGTH,BLENGTH)
          CBMAXWIDTH=MAX(CBMAXWIDTH,BWIDTH)

      ENDIF

10    IF (BLENGTH.GE.175..AND.BLENGTH.LE.194..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
```

```
            CCCOUNT=CCCOUNT +1
            CCLENGTH=CCLENGTH+BLENGTH
            CCWIDTH=CCWIDTH+BWIDTH
            CCCAPACITY=CCCAPACITY+CAPACITY
            CCMAXCAPACITY=MAX(CCMAXCAPACITY,CAPACITY)
            CCMAXLENGTH=MAX(CCMAXLENGTH,BLENGTH)
            CCMAXWIDTH=MAX(CCMAXWIDTH,BWIDTH)

      ENDIF

11    IF (BLENGTH.GE.175..AND.BLENGTH.LE.194..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            CDCOUNT=CDCOUNT +1
            CDLENGTH=CDLENGTH+BLENGTH
            CDWIDTH=CDWIDTH+BWIDTH
            CDCAPACITY=CDCAPACITY+CAPACITY
            CDMAXCAPACITY=MAX(CDMAXCAPACITY,CAPACITY)
            CDMAXLENGTH=MAX(CDMAXLENGTH,BLENGTH)
            CDMAXWIDTH=MAX(CDMAXWIDTH,BWIDTH)

      ENDIF

12    IF (BLENGTH.GE.195..AND.BLENGTH.LE.199..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            DBCOUNT=DBCOUNT +1
            DBLENGTH=DBLENGTH+BLENGTH
            DBWIDTH=DBWIDTH+BWIDTH
            DBCAPACITY=DBCAPACITY+CAPACITY
            DBMAXCAPACITY=MAX(DBMAXCAPACITY,CAPACITY)
            DBMAXLENGTH=MAX(DBMAXLENGTH,BLENGTH)
            DBMAXWIDTH=MAX(DBMAXWIDTH,BWIDTH)

      ENDIF

13    IF (BLENGTH.GE.195..AND.BLENGTH.LE.199..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            DCCOUNT=DCCOUNT +1
            DCLENGTH=DCLENGTH+BLENGTH
            DCWIDTH=DCWIDTH+BWIDTH
            DCCAPACITY=DCCAPACITY+CAPACITY
            DCMAXCAPACITY=MAX(DCMAXCAPACITY,CAPACITY)
            DCMAXLENGTH=MAX(DCMAXLENGTH,BLENGTH)
            DCMAXWIDTH=MAX(DCMAXWIDTH,BWIDTH)

      ENDIF

14    IF (BLENGTH.GE.195..AND.BLENGTH.LE.199..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            DDCOUNT=DDCOUNT +1
```

```
            DDLENGTH=DDLENGTH+BLENGTH
            DDWIDTH=DDWIDTH+BWIDTH
            DDCAPACITY=DDCAPACITY+CAPACITY
            DDMAXCAPACITY=MAX(DDMAXCAPACITY,CAPACITY)
            DDMAXLENGTH=MAX(DDMAXLENGTH,BLENGTH)
            DDMAXWIDTH=MAX(DDMAXWIDTH,BWIDTH)

      ENDIF

15    IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GT.10..AND.
     1    BWIDTH.LT.26. .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            EACOUNT=EACOUNT +1
            EALENGTH=EALENGTH+BLENGTH
            EAWIDTH=EAWIDTH+BWIDTH
            EACAPACITY=EACAPACITY+CAPACITY
            EAMAXCAPACITY=MAX(EAMAXCAPACITY,CAPACITY)
            EAMAXLENGTH=MAX(EAMAXLENGTH,BLENGTH)
            EAMAXWIDTH=MAX(EAMAXWIDTH,BWIDTH)

      ENDIF

16    IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            EBCOUNT=EBCOUNT +1
            EBLENGTH=EBLENGTH+BLENGTH
            EBWIDTH=EBWIDTH+BWIDTH
            EBCAPACITY=EBCAPACITY+CAPACITY
            EBMAXCAPACITY=MAX(EBMAXCAPACITY,CAPACITY)
            EBMAXLENGTH=MAX(EBMAXLENGTH,BLENGTH)
            EBMAXWIDTH=MAX(EBMAXWIDTH,BWIDTH)

      ENDIF

17    IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            ECCOUNT=ECCOUNT +1
            ECLENGTH=ECLENGTH+BLENGTH
            ECWIDTH=ECWIDTH+BWIDTH
            ECCAPACITY=ECCAPACITY+CAPACITY
            ECMAXCAPACITY=MAX(ECMAXCAPACITY,CAPACITY)
            ECMAXLENGTH=MAX(ECMAXLENGTH,BLENGTH)
            ECMAXWIDTH=MAX(ECMAXWIDTH,BWIDTH)

      ENDIF

18    IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            EDCOUNT=EDCOUNT +1
            EDLENGTH=EDLENGTH+BLENGTH
```

```
            EDWIDTH=EDWIDTH+BWIDTH
            EDCAPACITY=EDCAPACITY+CAPACITY
            EDMAXCAPACITY=MAX(EDMAXCAPACITY,CAPACITY)
            EDMAXLENGTH=MAX(EDMAXLENGTH,BLENGTH)
            EDMAXWIDTH=MAX(EDMAXWIDTH,BWIDTH)

         ENDIF

19       IF (BLENGTH.GE.260..AND.BLENGTH.LE.289..AND.BWIDTH.GE.35.0.AND.
        1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
        2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
        3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            FCCOUNT=FCCOUNT +1
            FCLENGTH=FCLENGTH+BLENGTH
            FCWIDTH=FCWIDTH+BWIDTH
            FCCAPACITY=FCCAPACITY+CAPACITY
            FCMAXCAPACITY=MAX(FCMAXCAPACITY,CAPACITY)
            FCMAXLENGTH=MAX(FCMAXLENGTH,BLENGTH)
            FCMAXWIDTH=MAX(FCMAXWIDTH,BWIDTH)

         ENDIF

20       IF (BLENGTH.GE.260..AND.BLENGTH.LE.289..AND.BWIDTH.GT.54.0.AND.
        1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
        2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
        3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            FDCOUNT=FDCOUNT +1
            FDLENGTH=FDLENGTH+BLENGTH
            FDWIDTH=FDWIDTH+BWIDTH
            FDCAPACITY=FDCAPACITY+CAPACITY
            FDMAXCAPACITY=MAX(FDMAXCAPACITY,CAPACITY)
            FDMAXLENGTH=MAX(FDMAXLENGTH,BLENGTH)
            FDMAXWIDTH=MAX(FDMAXWIDTH,BWIDTH)

         ENDIF

21       IF (BLENGTH.GE.290..AND.BLENGTH.LE.300..AND.BWIDTH.GE.35.0.AND.
        1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
        2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
        3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            GCCOUNT=GCCOUNT +1
            GCLENGTH=GCLENGTH+BLENGTH
            GCWIDTH=GCWIDTH+BWIDTH
            GCCAPACITY=GCCAPACITY+CAPACITY
            GCMAXCAPACITY=MAX(GCMAXCAPACITY,CAPACITY)
            GCMAXLENGTH=MAX(GCMAXLENGTH,BLENGTH)
            GCMAXWIDTH=MAX(GCMAXWIDTH,BWIDTH)

         ENDIF

22       IF (BLENGTH.GE.290..AND.BLENGTH.LE.300..AND.BWIDTH.GT.54.0.AND.
        1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
        2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
        3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            GDCOUNT=GDCOUNT +1
            GDLENGTH=GDLENGTH+BLENGTH
            GDWIDTH=GDWIDTH+BWIDTH
```

```
            GDCAPACITY=GDCAPACITY+CAPACITY
            GDMAXCAPACITY=MAX(GDMAXCAPACITY,CAPACITY)
            GDMAXLENGTH=MAX(GDMAXLENGTH,BLENGTH)
            GDMAXWIDTH=MAX(GDMAXWIDTH,BWIDTH)

         ENDIF

23       IF (BLENGTH.GT.300.0 .AND.BWIDTH.GE.26..AND.BWIDTH.LT.35.0 .AND.
        1     IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
        2    CAPACITY.GT.10.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
        3     .GT. 0.0 .AND. LIGHTDRAFT .GT. 0.0) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            HBCOUNT=HBCOUNT +1
            HBLENGTH=HBLENGTH+BLENGTH
            HBWIDTH=HBWIDTH+BWIDTH
            HBCAPACITY=HBCAPACITY+CAPACITY
            HBMAXCAPACITY=MAX(HBMAXCAPACITY,CAPACITY)
            HBMAXLENGTH=MAX(HBMAXLENGTH,BLENGTH)
            HBMAXWIDTH=MAX(HBMAXWIDTH,BWIDTH)

         ENDIF

24       IF (BLENGTH.GT.300.0 .AND. BLENGTH .LT. 500.0 .AND.BWIDTH.GE.35.0
        1      .AND.BWIDTH.LE.54.0 .AND.
        2     IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
        3     CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
        3     .GT.LIGHTDRAFT) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            HCCOUNT=HCCOUNT +1
            HCLENGTH=HCLENGTH+BLENGTH
            HCWIDTH=HCWIDTH+BWIDTH
            HCCAPACITY=HCCAPACITY+CAPACITY
            HCMAXCAPACITY=MAX(HCMAXCAPACITY,CAPACITY)
            HCMAXLENGTH=MAX(HCMAXLENGTH,BLENGTH)
            HCMAXWIDTH=MAX(HCMAXWIDTH,BWIDTH)

         ENDIF

25       IF (BLENGTH.GT.300. .AND.BWIDTH.GT.54.0 .AND. IAREA.EQ.4 .AND.
        1     BWIDTH.LE.79 .AND. LOADDRAFT.LT.15.2 .AND.
        2     CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 ) THEN

CAPACITY=CAPACITY+(LIGHTDRAFT/(LOADDRAFT-LIGHTDRAFT))*CAPACITY
            HDCOUNT=HDCOUNT +1
            HDLENGTH=HDLENGTH+BLENGTH
            HDWIDTH=HDWIDTH+BWIDTH
            HDCAPACITY=HDCAPACITY+CAPACITY
            HDMAXCAPACITY=MAX(HDMAXCAPACITY,CAPACITY)
            HDMAXLENGTH=MAX(HDMAXLENGTH,BLENGTH)
            HDMAXWIDTH=MAX(HDMAXWIDTH,BWIDTH)

         ENDIF

100      END DO

101      AACAPACITY=AACAPACITY/AACOUNT
         ABCAPACITY=ABCAPACITY/ABCOUNT
         ACCAPACITY=ACCAPACITY/ACCOUNT
         ADCAPACITY=ADCAPACITY/ADCOUNT

         BACAPACITY=BACAPACITY/BACOUNT
         BBCAPACITY=BBCAPACITY/BBCOUNT
         BCCAPACITY=BCCAPACITY/BCCOUNT
```

```
BDCAPACITY=BDCAPACITY/BDCOUNT

CBCAPACITY=CBCAPACITY/CBCOUNT
CCCAPACITY=CCCAPACITY/CCCOUNT
CDCAPACITY=CDCAPACITY/CDCOUNT

DBCAPACITY=DBCAPACITY/DBCOUNT
DCCAPACITY=DCCAPACITY/DCCOUNT
DDCAPACITY=DDCAPACITY/DDCOUNT

EACAPACITY=EACAPACITY/EACOUNT
EBCAPACITY=EBCAPACITY/EBCOUNT
ECCAPACITY=ECCAPACITY/ECCOUNT
EDCAPACITY=EDCAPACITY/EDCOUNT

FCCAPACITY=FCCAPACITY/FCCOUNT
FDCAPACITY=FDCAPACITY/FDCOUNT

GCCAPACITY=GCCAPACITY/GCCOUNT
GDCAPACITY=GDCAPACITY/GDCOUNT

HBCAPACITY=HBCAPACITY/HBCOUNT
HCCAPACITY=HCCAPACITY/HCCOUNT
HDCAPACITY=HDCAPACITY/HDCOUNT

AAWIDTH=AAWIDTH/AACOUNT
ABWIDTH=ABWIDTH/ABCOUNT
ACWIDTH=ACWIDTH/ACCOUNT
ADWIDTH=ADWIDTH/ADCOUNT

BAWIDTH=BAWIDTH/BACOUNT
BBWIDTH=BBWIDTH/BBCOUNT
BCWIDTH=BCWIDTH/BCCOUNT
BDWIDTH=BDWIDTH/BDCOUNT

CBWIDTH=CBWIDTH/CBCOUNT
CCWIDTH=CCWIDTH/CCCOUNT
CDWIDTH=CDWIDTH/CDCOUNT

DBWIDTH=DBWIDTH/DBCOUNT
DCWIDTH=DCWIDTH/DCCOUNT
DDWIDTH=DDWIDTH/DDCOUNT

EAWIDTH=EAWIDTH/EACOUNT
EBWIDTH=EBWIDTH/EBCOUNT
ECWIDTH=ECWIDTH/ECCOUNT
EDWIDTH=EDWIDTH/EDCOUNT

FCWIDTH=FCWIDTH/FCCOUNT
FDWIDTH=FDWIDTH/FDCOUNT

GCWIDTH=GCWIDTH/GCCOUNT
GDWIDTH=GDWIDTH/GDCOUNT

HBWIDTH=HBWIDTH/HBCOUNT
HCWIDTH=HCWIDTH/HCCOUNT
HDWIDTH=HDWIDTH/HDCOUNT

AALENGTH=AALENGTH/AACOUNT
ABLENGTH=ABLENGTH/ABCOUNT
ACLENGTH=ACLENGTH/ACCOUNT
ADLENGTH=ADLENGTH/ADCOUNT

BALENGTH=BALENGTH/BACOUNT
BBLENGTH=BBLENGTH/BBCOUNT
```

```
      BCLENGTH=BCLENGTH/BCCOUNT
      BDLENGTH=BDLENGTH/BDCOUNT

      CBLENGTH=CBLENGTH/CBCOUNT
      CCLENGTH=CCLENGTH/CCCOUNT
      CDLENGTH=CDLENGTH/CDCOUNT

      DBLENGTH=DBLENGTH/DBCOUNT
      DCLENGTH=DCLENGTH/DCCOUNT
      DDLENGTH=DDLENGTH/DDCOUNT

      EALENGTH=EALENGTH/EACOUNT
      EBLENGTH=EBLENGTH/EBCOUNT
      ECLENGTH=ECLENGTH/ECCOUNT
      EDLENGTH=EDLENGTH/EDCOUNT

      FCLENGTH=FCLENGTH/FCCOUNT
      FDLENGTH=FDLENGTH/FDCOUNT

      GCLENGTH=GCLENGTH/GCCOUNT
      GDLENGTH=GDLENGTH/GDCOUNT

      HBLENGTH=HBLENGTH/HBCOUNT
      HCLENGTH=HCLENGTH/HCCOUNT
      HDLENGTH=HDLENGTH/HDCOUNT

                  WRITE(2,*)    "******************CATEGORY    AVERAGE
VALUES************"
      WRITE(2,*)"CATEG.     ","COUNT          ","CAPACITY     ",
     1          "LENGTH         ","WIDTH"

      WRITE(2,*)"AA",AACOUNT,AACAPACITY,AALENGTH,AAWIDTH
      WRITE(2,*)"AB",ABCOUNT,ABCAPACITY,ABLENGTH,ABWIDTH
      WRITE(2,*)"AC",ACCOUNT,ACCAPACITY,ACLENGTH,ACWIDTH
      WRITE(2,*)"AD",ADCOUNT,ADCAPACITY,ADLENGTH,ADWIDTH

      WRITE(2,*)"BA",BACOUNT,BACAPACITY,BALENGTH,BAWIDTH
      WRITE(2,*)"BB",BBCOUNT,BBCAPACITY,BBLENGTH,BBWIDTH
      WRITE(2,*)"BC",BCCOUNT,BCCAPACITY,BCLENGTH,BCWIDTH
      WRITE(2,*)"BD",BDCOUNT,BDCAPACITY,BDLENGTH,BDWIDTH

      WRITE(2,*)"CB",CBCOUNT,CBCAPACITY,CBLENGTH,CBWIDTH
      WRITE(2,*)"CC",CCCOUNT,CCCAPACITY,CCLENGTH,CCWIDTH
      WRITE(2,*)"CD",CDCOUNT,CDCAPACITY,CDLENGTH,CDWIDTH

      WRITE(2,*)"DB",DBCOUNT,DBCAPACITY,DBLENGTH,DBWIDTH
      WRITE(2,*)"DC",DCCOUNT,DCCAPACITY,DCLENGTH,DCWIDTH
      WRITE(2,*)"DD",DDCOUNT,DDCAPACITY,DDLENGTH,DDWIDTH

      WRITE(2,*)"EA",EACOUNT,EACAPACITY,EALENGTH,EAWIDTH
      WRITE(2,*)"EB",EBCOUNT,EBCAPACITY,EBLENGTH,EBWIDTH
      WRITE(2,*)"EC",ECCOUNT,ECCAPACITY,ECLENGTH,ECWIDTH
      WRITE(2,*)"ED",EDCOUNT,EDCAPACITY,EDLENGTH,EDWIDTH

      WRITE(2,*)"FC",FCCOUNT,FCCAPACITY,FCLENGTH,FCWIDTH
      WRITE(2,*)"FD",FDCOUNT,FDCAPACITY,FDLENGTH,FDWIDTH

      WRITE(2,*)"GC",GCCOUNT,GCCAPACITY,GCLENGTH,GCWIDTH
      WRITE(2,*)"GD",GDCOUNT,GDCAPACITY,GDLENGTH,GDWIDTH

      WRITE(2,*)"HB",HBCOUNT,HBCAPACITY,HBLENGTH,HBWIDTH
      WRITE(2,*)"HC",HCCOUNT,HCCAPACITY,HCLENGTH,HCWIDTH
      WRITE(2,*)"HD",HDCOUNT,HDCAPACITY,HDLENGTH,HDWIDTH
```

```
      REWIND(1)
C
C     CALCULATE STANDARD DEVIATIONS OF BARGE CHARACTERISTICS
C
      DO WHILE(.NOT.EOF(1))

      READ (1,*) BLENGTH,BWIDTH,CAPACITY,LOADDRAFT,LIGHTDRAFT,IAREA

      IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GT.10..AND.
     1    BWIDTH.LT.26. .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.10.0 .AND. LOADDRAFT
     3    .GT. 0.0 .AND. LIGHTDRAFT .GT. 0.0) THEN
          AASDCAPACITY=AASDCAPACITY+(CAPACITY-AACAPACITY)**2
          AASDLENGTH=AASDLENGTH+(BLENGTH-AALENGTH)**2
          AASDWIDTH=AASDWIDTH+(BWIDTH-AAWIDTH)**2

      ENDIF

      IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 4000.0 .AND. LOADDRAFT
     3    .GT. 0.0 .AND. LIGHTDRAFT .GT. 0.0) THEN
          ABSDCAPACITY=ABSDCAPACITY+(CAPACITY-ABCAPACITY)**2
          ABSDLENGTH=ABSDLENGTH+(BLENGTH-ABLENGTH)**2
          ABSDWIDTH=ABSDWIDTH+(BWIDTH-ABWIDTH)**2

      ENDIF

      IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 5000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN
          ACSDCAPACITY=ACSDCAPACITY+(CAPACITY-ACCAPACITY)**2
          ACSDLENGTH=ACSDLENGTH+(BLENGTH-ACLENGTH)**2
          ACSDWIDTH=ACSDWIDTH+(BWIDTH-ACWIDTH)**2

      ENDIF

      IF (BLENGTH.GT.10..AND.BLENGTH.LT.100..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN
          ADSDCAPACITY=ADSDCAPACITY+(CAPACITY-ADCAPACITY)**2
          ADSDLENGTH=ADSDLENGTH+(BLENGTH-ADLENGTH)**2
          ADSDWIDTH=ADSDWIDTH+(BWIDTH-ADWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GT.10..AND.
     1    BWIDTH.LT.26. .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN
          BASDCAPACITY=BASDCAPACITY+(CAPACITY-BACAPACITY)**2
          BASDLENGTH=BASDLENGTH+(BLENGTH-BALENGTH)**2
          BASDWIDTH=BASDWIDTH+(BWIDTH-BAWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3    .GT.LIGHTDRAFT) THEN
```

```
      BBSDCAPACITY=BBSDCAPACITY+(CAPACITY-BBCAPACITY)**2
      BBSDLENGTH=BBSDLENGTH+(BLENGTH-BBLENGTH)**2
      BBSDWIDTH=BBSDWIDTH+(BWIDTH-BBWIDTH)**2

 ENDIF

 IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GE.35.0.AND.
1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
3     .GT.LIGHTDRAFT) THEN
      BCSDCAPACITY=BCSDCAPACITY+(CAPACITY-BCCAPACITY)**2
      BCSDLENGTH=BCSDLENGTH+(BLENGTH-BCLENGTH)**2
      BCSDWIDTH=BCSDWIDTH+(BWIDTH-BCWIDTH)**2

 ENDIF

 IF (BLENGTH.GE.100..AND.BLENGTH.LE.174..AND.BWIDTH.GT.54.0.AND.
1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
3     .GT.LIGHTDRAFT) THEN
      BDSDCAPACITY=BDSDCAPACITY+(CAPACITY-BDCAPACITY)**2
      BDSDLENGTH=BDSDLENGTH+(BLENGTH-BDLENGTH)**2
      BDSDWIDTH=BDSDWIDTH+(BWIDTH-BDWIDTH)**2

 ENDIF

 IF (BLENGTH.GE.175..AND.BLENGTH.LE.194..AND.BWIDTH.GE.26..AND.
1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
3     .GT.LIGHTDRAFT) THEN
      CBSDCAPACITY=CBSDCAPACITY+(CAPACITY-CBCAPACITY)**2
      CBSDLENGTH=CBSDLENGTH+(BLENGTH-CBLENGTH)**2
      CBSDWIDTH=CBSDWIDTH+(BWIDTH-CBWIDTH)**2

 ENDIF

 IF (BLENGTH.GE.175..AND.BLENGTH.LE.194..AND.BWIDTH.GE.35.0.AND.
1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
3     .GT.LIGHTDRAFT) THEN
      CCSDCAPACITY=CCSDCAPACITY+(CAPACITY-CCCAPACITY)**2
      CCSDLENGTH=CCSDLENGTH+(BLENGTH-CCLENGTH)**2
      CCSDWIDTH=CCSDWIDTH+(BWIDTH-CCWIDTH)**2

 ENDIF

 IF (BLENGTH.GE.175..AND.BLENGTH.LE.194..AND.BWIDTH.GT.54.0.AND.
1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
3     .GT.LIGHTDRAFT) THEN
      CDSDCAPACITY=CDSDCAPACITY+(CAPACITY-CDCAPACITY)**2
      CDSDLENGTH=CDSDLENGTH+(BLENGTH-CDLENGTH)**2
      CDSDWIDTH=CDSDWIDTH+(BWIDTH-CDWIDTH)**2

 ENDIF

 IF (BLENGTH.GE.195..AND.BLENGTH.LE.199..AND.BWIDTH.GE.26..AND.
1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
3     .GT.LIGHTDRAFT) THEN
      DBSDCAPACITY=DBSDCAPACITY+(CAPACITY-DBCAPACITY)**2
      DBSDLENGTH=DBSDLENGTH+(BLENGTH-DBLENGTH)**2
      DBSDWIDTH=DBSDWIDTH+(BWIDTH-DBWIDTH)**2

 ENDIF
```

```
      IF (BLENGTH.GE.195..AND.BLENGTH.LE.199..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
           DCSDCAPACITY=DCSDCAPACITY+(CAPACITY-DCCAPACITY)**2
           DCSDLENGTH=DCSDLENGTH+(BLENGTH-DCLENGTH)**2
           DCSDWIDTH=DCSDWIDTH+(BWIDTH-DCWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.195..AND.BLENGTH.LE.199..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
           DDSDCAPACITY=DDSDCAPACITY+(CAPACITY-DDCAPACITY)**2
           DDSDLENGTH=DDSDLENGTH+(BLENGTH-DDLENGTH)**2
           DDSDWIDTH=DDSDWIDTH+(BWIDTH-DDWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GT.10..AND.
     1    BWIDTH.LT.26. .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
           EASDCAPACITY=EASDCAPACITY+(CAPACITY-EACAPACITY)**2
           EASDLENGTH=EASDLENGTH+(BLENGTH-EALENGTH)**2
           EASDWIDTH=EASDWIDTH+(BWIDTH-EAWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GE.26..AND.
     1    BWIDTH.LT.35.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
           EBSDCAPACITY=EBSDCAPACITY+(CAPACITY-EBCAPACITY)**2
           EBSDLENGTH=EBSDLENGTH+(BLENGTH-EBLENGTH)**2
           EBSDWIDTH=EBSDWIDTH+(BWIDTH-EBWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
           ECSDCAPACITY=ECSDCAPACITY+(CAPACITY-ECCAPACITY)**2
           ECSDLENGTH=ECSDLENGTH+(BLENGTH-ECLENGTH)**2
           ECSDWIDTH=ECSDWIDTH+(BWIDTH-ECWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.200..AND.BLENGTH.LE.259..AND.BWIDTH.GT.54.0.AND.
     1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
           EDSDCAPACITY=EDSDCAPACITY+(CAPACITY-EDCAPACITY)**2
           EDSDLENGTH=EDSDLENGTH+(BLENGTH-EDLENGTH)**2
           EDSDWIDTH=EDSDWIDTH+(BWIDTH-EDWIDTH)**2

      ENDIF

      IF (BLENGTH.GE.260..AND.BLENGTH.LE.289..AND.BWIDTH.GE.35.0.AND.
     1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
     2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
     3     .GT.LIGHTDRAFT) THEN
```

```
      FCSDCAPACITY=FCSDCAPACITY+(CAPACITY-FCCAPACITY)**2
      FCSDLENGTH=FCSDLENGTH+(BLENGTH-FCLENGTH)**2
      FCSDWIDTH=FCSDWIDTH+(BWIDTH-FCWIDTH)**2

     ENDIF

     IF (BLENGTH.GE.260..AND.BLENGTH.LE.289..AND.BWIDTH.GT.54.0.AND.
    1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
    2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
    3     .GT.LIGHTDRAFT) THEN
      FDSDCAPACITY=FDSDCAPACITY+(CAPACITY-FDCAPACITY)**2
      FDSDLENGTH=FDSDLENGTH+(BLENGTH-FDLENGTH)**2
      FDSDWIDTH=FDSDWIDTH+(BWIDTH-FDWIDTH)**2

     ENDIF

     IF (BLENGTH.GE.290..AND.BLENGTH.LE.300..AND.BWIDTH.GE.35.0.AND.
    1    BWIDTH.LE.54.0 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
    2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
    3     .GT.LIGHTDRAFT) THEN
      GCSDCAPACITY=GCSDCAPACITY+(CAPACITY-GCCAPACITY)**2
      GCSDLENGTH=GCSDLENGTH+(BLENGTH-GCLENGTH)**2
      GCSDWIDTH=GCSDWIDTH+(BWIDTH-GCWIDTH)**2

     ENDIF

     IF (BLENGTH.GE.290..AND.BLENGTH.LE.300..AND.BWIDTH.GT.54.0.AND.
    1    BWIDTH.LE.79 .AND. IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
    2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
    3     .GT.LIGHTDRAFT) THEN
      GDSDCAPACITY=GDSDCAPACITY+(CAPACITY-GDCAPACITY)**2
      GDSDLENGTH=GDSDLENGTH+(BLENGTH-GDLENGTH)**2
      GDSDWIDTH=GDSDWIDTH+(BWIDTH-GDWIDTH)**2

     ENDIF

     IF (BLENGTH.GT.300. .AND.BWIDTH.GE.26..AND.BWIDTH.LT.35.0 .AND.
    1    IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
    2    CAPACITY.GT.10.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
    3     .GT. 0.0 .AND. LIGHTDRAFT .GT. 0.0) THEN
      HBSDCAPACITY=HBSDCAPACITY+(CAPACITY-HBCAPACITY)**2
      HBSDLENGTH=HBSDLENGTH+(BLENGTH-HBLENGTH)**2
      HBSDWIDTH=HBSDWIDTH+(BWIDTH-HBWIDTH)**2

     ENDIF

     IF (BLENGTH.GT.300.0 .AND. BLENGTH .LT. 500.0
    1      .AND.BWIDTH.GE.35.0.AND.BWIDTH.LE.54.0 .AND.
    2    IAREA.EQ.4 .AND. LOADDRAFT.LT.15.2 .AND.
    3    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
    3     .GT.LIGHTDRAFT) THEN
      HCSDCAPACITY=HCSDCAPACITY+(CAPACITY-HCCAPACITY)**2
      HCSDLENGTH=HCSDLENGTH+(BLENGTH-HCLENGTH)**2
      HCSDWIDTH=HCSDWIDTH+(BWIDTH-HCWIDTH)**2

     ENDIF

     IF (BLENGTH.GT.300. .AND.BWIDTH.GT.54.0 .AND. IAREA.EQ.4 .AND.
    1    BWIDTH.LE.79 .AND. LOADDRAFT.LT.15.2 .AND.
    2    CAPACITY.GT.99.0 .AND. CAPACITY .LT. 10000.0 .AND. LOADDRAFT
    3     .GT.LIGHTDRAFT) THEN
      HDSDCAPACITY=HDSDCAPACITY+(CAPACITY-HDCAPACITY)**2
      HDSDLENGTH=HDSDLENGTH+(BLENGTH-HDLENGTH)**2
      HDSDWIDTH=HDSDWIDTH+(BWIDTH-HDWIDTH)**2
```

```
      ENDIF
200   END DO
C
C          NOTE: CATEGORIES WITH SMALL COUNTS DO NOT INCLUDE THE
C                TWO STANDARD DEVIATIONS
C
201   AACAPACITY=AACAPACITY+2*SQRT(AASDCAPACITY/(AACOUNT-1))

      ABCAPACITY=ABCAPACITY+2*SQRT(ABSDCAPACITY/(ABCOUNT-1))
      ACCAPACITY=ACCAPACITY+2*SQRT(ACSDCAPACITY/(ACCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE ADCOUNT=1
C
C     ADCAPACITY=ADCAPACITY+2*SQRT(ADSDCAPACITY/(ADCOUNT-1))

      BACAPACITY=BACAPACITY+2*SQRT(BASDCAPACITY/(BACOUNT-1))
      BBCAPACITY=BBCAPACITY+2*SQRT(BBSDCAPACITY/(BBCOUNT-1))
      BCCAPACITY=BCCAPACITY+2*SQRT(BCSDCAPACITY/(BCCOUNT-1))

      BDCAPACITY=BDCAPACITY+2*SQRT(BDSDCAPACITY/(BDCOUNT-1))

      CBCAPACITY=CBCAPACITY+2*SQRT(CBSDCAPACITY/(CBCOUNT-1))
      CCCAPACITY=CCCAPACITY+2*SQRT(CCSDCAPACITY/(CCCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE CDCOUNT=1
C
C     CDCAPACITY=CDCAPACITY+2*SQRT(CDSDCAPACITY/(CDCOUNT-1))

      DBCAPACITY=DBCAPACITY+2*SQRT(DBSDCAPACITY/(DBCOUNT-1))
      DCCAPACITY=DCCAPACITY+2*SQRT(DCSDCAPACITY/(DCCOUNT-1))

C
C     NEGLECT FOLLOWING LINE SINCE DDCOUNT=1
C
C     DDCAPACITY=DDCAPACITY+2*SQRT(DDSDCAPACITY/(DDCOUNT-1))

      EACAPACITY=EACAPACITY+2*SQRT(EASDCAPACITY/(EACOUNT-1))

      EBCAPACITY=EBCAPACITY+2*SQRT(EBSDCAPACITY/(EBCOUNT-1))
      ECCAPACITY=ECCAPACITY+2*SQRT(ECSDCAPACITY/(ECCOUNT-1))

      EDCAPACITY=EDCAPACITY+2*SQRT(EDSDCAPACITY/(EDCOUNT-1))

      FCCAPACITY=FCCAPACITY+2*SQRT(FCSDCAPACITY/(FCCOUNT-1))

      FDCAPACITY=FDCAPACITY+2*SQRT(FDSDCAPACITY/(FDCOUNT-1))

      GCCAPACITY=GCCAPACITY+2*SQRT(GCSDCAPACITY/(GCCOUNT-1))

      GDCAPACITY=GDCAPACITY+2*SQRT(GDSDCAPACITY/(GDCOUNT-1))

C
C     NEGLECT FOLLOWING LINE SINCE HBCOUNT=1
C
C     HBCAPACITY=HBCAPACITY+2*SQRT(HBSDCAPACITY/(HBCOUNT-1))
      HCCAPACITY=HCCAPACITY+2*SQRT(HCSDCAPACITY/(HCCOUNT-1))

      HDCAPACITY=HDCAPACITY+2*SQRT(HDSDCAPACITY/(HDCOUNT-1))

      AAWIDTH=AAWIDTH+2*SQRT(AASDWIDTH/(AACOUNT-1))
      ABWIDTH=ABWIDTH+2*SQRT(ABSDWIDTH/(ABCOUNT-1))
```

```
      ACWIDTH=ACWIDTH+2*SQRT(ACSDWIDTH/(ACCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE ADCOUNT=1
C
C     ADWIDTH=ADWIDTH+2*SQRT(ADSDWIDTH/(ADCOUNT-1))

      BAWIDTH=BAWIDTH+2*SQRT(BASDWIDTH/(BACOUNT-1))
      BBWIDTH=BBWIDTH+2*SQRT(BBSDWIDTH/(BBCOUNT-1))
      BCWIDTH=BCWIDTH+2*SQRT(BCSDWIDTH/(BCCOUNT-1))
      BDWIDTH=BDWIDTH+2*SQRT(BDSDWIDTH/(BDCOUNT-1))

      CBWIDTH=CBWIDTH+2*SQRT(CBSDWIDTH/(CBCOUNT-1))
      CCWIDTH=CCWIDTH+2*SQRT(CCSDWIDTH/(CCCOUNT-1))

C
C     NEGLECT FOLLOWING LINE SINCE CDCOUNT=1
C
C     CDWIDTH=CDWIDTH+2*SQRT(CDSDWIDTH/(CDCOUNT-1))

      DBWIDTH=DBWIDTH+2*SQRT(DBSDWIDTH/(DBCOUNT-1))
      DCWIDTH=DCWIDTH+2*SQRT(DCSDWIDTH/(DCCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE DDCOUNT=1
C
C     DDWIDTH=DDWIDTH+2*SQRT(DDSDWIDTH/(DDCOUNT-1))

      EAWIDTH=EAWIDTH+2*SQRT(EASDWIDTH/(EACOUNT-1))
      EBWIDTH=EBWIDTH+2*SQRT(EBSDWIDTH/(EBCOUNT-1))
      ECWIDTH=ECWIDTH+2*SQRT(ECSDWIDTH/(ECCOUNT-1))
      EDWIDTH=EDWIDTH+2*SQRT(EDSDWIDTH/(EDCOUNT-1))

      FCWIDTH=FCWIDTH+2*SQRT(FCSDWIDTH/(FCCOUNT-1))
      FDWIDTH=FDWIDTH+2*SQRT(FDSDWIDTH/(FDCOUNT-1))

      GCWIDTH=GCWIDTH+2*SQRT(GCSDWIDTH/(GCCOUNT-1))
      GDWIDTH=GDWIDTH+2*SQRT(GDSDWIDTH/(GDCOUNT-1))

C
C     NEGLECT FOLLOWING LINE SINCE HBCOUNT=1
C
C     HBWIDTH=HBWIDTH+2*SQRT(HBSDWIDTH/(HBCOUNT-1))
      HCWIDTH=HCWIDTH+2*SQRT(HCSDWIDTH/(HCCOUNT-1))
      HDWIDTH=HDWIDTH+2*SQRT(HDSDWIDTH/(HDCOUNT-1))

      AALENGTH=AALENGTH+2*SQRT(AASDLENGTH/(AACOUNT-1))
      ABLENGTH=ABLENGTH+2*SQRT(ABSDLENGTH/(ABCOUNT-1))
      ACLENGTH=ACLENGTH+2*SQRT(ACSDLENGTH/(ACCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE ADCOUNT=1
C
C     ADLENGTH=ADLENGTH+2*SQRT(ADSDLENGTH/(ADCOUNT-1))

      BALENGTH=BALENGTH+2*SQRT(BASDLENGTH/(BACOUNT-1))
      BBLENGTH=BBLENGTH+2*SQRT(BBSDLENGTH/(BBCOUNT-1))
      BCLENGTH=BCLENGTH+2*SQRT(BCSDLENGTH/(BCCOUNT-1))
      BDLENGTH=BDLENGTH+2*SQRT(BDSDLENGTH/(BDCOUNT-1))

      CBLENGTH=CBLENGTH+2*SQRT(CBSDLENGTH/(CBCOUNT-1))
      CCLENGTH=CCLENGTH+2*SQRT(CCSDLENGTH/(CCCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE CDCOUNT=1
C
C     CDLENGTH=CDLENGTH+2*SQRT(CDSDLENGTH/(CDCOUNT-1))

      DBLENGTH=DBLENGTH+2*SQRT(DBSDLENGTH/(DBCOUNT-1))
```

```
      DCLENGTH=DCLENGTH+2*SQRT(DCSDLENGTH/(DCCOUNT-1))
C
C     NEGLECT FOLLOWING LINE SINCE DDCOUNT=1
C
C     DDLENGTH=DDLENGTH+2*SQRT(DDSDLENGTH/(DDCOUNT-1))

      EALENGTH=EALENGTH+2*SQRT(EASDLENGTH/(EACOUNT-1))
      EBLENGTH=EBLENGTH+2*SQRT(EBSDLENGTH/(EBCOUNT-1))
      ECLENGTH=ECLENGTH+2*SQRT(ECSDLENGTH/(ECCOUNT-1))
      EDLENGTH=EDLENGTH+2*SQRT(EDSDLENGTH/(EDCOUNT-1))

      FCLENGTH=FCLENGTH+2*SQRT(FCSDLENGTH/(FCCOUNT-1))
      FDLENGTH=FDLENGTH+2*SQRT(FDSDLENGTH/(FDCOUNT-1))

      GCLENGTH=GCLENGTH+2*SQRT(GCSDLENGTH/(GCCOUNT-1))
      GDLENGTH=GDLENGTH+2*SQRT(GDSDLENGTH/(GDCOUNT-1))

C
C     NEGLECT FOLLOWING LINE SINCE HBCOUNT=1
C
C     HBLENGTH=HBLENGTH+2*SQRT(HBSDLENGTH/(HBCOUNT-1))
      HCLENGTH=HCLENGTH+2*SQRT(HCSDLENGTH/(HCCOUNT-1))
      HDLENGTH=HDLENGTH+2*SQRT(HDSDLENGTH/(HDCOUNT-1))

C
C     FIND MINIMUM OF AVERAGE PLUS 2 STANDARD DEVIATIONS OR
C     MAXIMUM CHARACTERISTIC VALUE
C
      AACAPACITY=MIN(AACAPACITY,AAMAXCAPACITY)
      ABCAPACITY=MIN(ABCAPACITY,ABMAXCAPACITY)
      ACCAPACITY=MIN(ACCAPACITY,ACMAXCAPACITY)
C     ADCAPACITY=MIN(ADCAPACITY,ADMAXCAPACITY)

      BACAPACITY=MIN(BACAPACITY,BAMAXCAPACITY)
      BBCAPACITY=MIN(BBCAPACITY,BBMAXCAPACITY)
      BCCAPACITY=MIN(BCCAPACITY,BCMAXCAPACITY)
      BDCAPACITY=MIN(BDCAPACITY,BDMAXCAPACITY)

      CBCAPACITY=MIN(CBCAPACITY,CBMAXCAPACITY)
      CCCAPACITY=MIN(CCCAPACITY,CCMAXCAPACITY)
      CDCAPACITY=MIN(CDCAPACITY,CDMAXCAPACITY)

      DBCAPACITY=MIN(DBCAPACITY,DBMAXCAPACITY)
      DCCAPACITY=MIN(DCCAPACITY,DCMAXCAPACITY)
C     DDCAPACITY=MIN(DDCAPACITY,DDMAXCAPACITY)

      EACAPACITY=MIN(EACAPACITY,EAMAXCAPACITY)
      EBCAPACITY=MIN(EBCAPACITY,EBMAXCAPACITY)
      ECCAPACITY=MIN(ECCAPACITY,ECMAXCAPACITY)
      EDCAPACITY=MIN(EDCAPACITY,EDMAXCAPACITY)

      FCCAPACITY=MIN(FCCAPACITY,FCMAXCAPACITY)
      FDCAPACITY=MIN(FDCAPACITY,FDMAXCAPACITY)

      GCCAPACITY=MIN(GCCAPACITY,GCMAXCAPACITY)
      GDCAPACITY=MIN(GDCAPACITY,GDMAXCAPACITY)

      HBCAPACITY=MIN(HBCAPACITY,HBMAXCAPACITY)
      HCCAPACITY=MIN(HCCAPACITY,HCMAXCAPACITY)
      HDCAPACITY=MIN(HDCAPACITY,HDMAXCAPACITY)


      AALENGTH=MIN(AALENGTH,AAMAXLENGTH)
      ABLENGTH=MIN(ABLENGTH,ABMAXLENGTH)
      ACLENGTH=MIN(ACLENGTH,ACMAXLENGTH)
```

```
C       ADLENGTH=MIN(ADLENGTH,ADMAXLENGTH)

        BALENGTH=MIN(BALENGTH,BAMAXLENGTH)
        BBLENGTH=MIN(BBLENGTH,BBMAXLENGTH)
        BCLENGTH=MIN(BCLENGTH,BCMAXLENGTH)
        BDLENGTH=MIN(BDLENGTH,BDMAXLENGTH)

        CBLENGTH=MIN(CBLENGTH,CBMAXLENGTH)
        CCLENGTH=MIN(CCLENGTH,CCMAXLENGTH)
        CDLENGTH=MIN(CDLENGTH,CDMAXLENGTH)

        DBLENGTH=MIN(DBLENGTH,DBMAXLENGTH)
        DCLENGTH=MIN(DCLENGTH,DCMAXLENGTH)
C       DDLENGTH=MIN(DDLENGTH,DDMAXLENGTH)

        EALENGTH=MIN(EALENGTH,EAMAXLENGTH)
        EBLENGTH=MIN(EBLENGTH,EBMAXLENGTH)
        ECLENGTH=MIN(ECLENGTH,ECMAXLENGTH)
        EDLENGTH=MIN(EDLENGTH,EDMAXLENGTH)

        FCLENGTH=MIN(FCLENGTH,FCMAXLENGTH)
        FDLENGTH=MIN(FDLENGTH,FDMAXLENGTH)

        GCLENGTH=MIN(GCLENGTH,GCMAXLENGTH)
        GDLENGTH=MIN(GDLENGTH,GDMAXLENGTH)

        HBLENGTH=MIN(HBLENGTH,HBMAXLENGTH)
        HCLENGTH=MIN(HCLENGTH,HCMAXLENGTH)
        HDLENGTH=MIN(HDLENGTH,HDMAXLENGTH)

        AAWIDTH=MIN(AAWIDTH,AAMAXWIDTH)
        ABWIDTH=MIN(ABWIDTH,ABMAXWIDTH)
        ACWIDTH=MIN(ACWIDTH,ACMAXWIDTH)
C       ADWIDTH=MIN(ADWIDTH,ADMAXWIDTH)

        BAWIDTH=MIN(BAWIDTH,BAMAXWIDTH)
        BBWIDTH=MIN(BBWIDTH,BBMAXWIDTH)
        BCWIDTH=MIN(BCWIDTH,BCMAXWIDTH)
        BDWIDTH=MIN(BDWIDTH,BDMAXWIDTH)

        CBWIDTH=MIN(CBWIDTH,CBMAXWIDTH)
        CCWIDTH=MIN(CCWIDTH,CCMAXWIDTH)
        CDWIDTH=MIN(CDWIDTH,CDMAXWIDTH)

        DBWIDTH=MIN(DBWIDTH,DBMAXWIDTH)
        DCWIDTH=MIN(DCWIDTH,DCMAXWIDTH)
C       DDWIDTH=MIN(DDWIDTH,DDMAXWIDTH)

        EAWIDTH=MIN(EAWIDTH,EAMAXWIDTH)
        EBWIDTH=MIN(EBWIDTH,EBMAXWIDTH)
        ECWIDTH=MIN(ECWIDTH,ECMAXWIDTH)
        EDWIDTH=MIN(EDWIDTH,EDMAXWIDTH)

        FCWIDTH=MIN(FCWIDTH,FCMAXWIDTH)
        FDWIDTH=MIN(FDWIDTH,FDMAXWIDTH)

        GCWIDTH=MIN(GCWIDTH,GCMAXWIDTH)
        GDWIDTH=MIN(GDWIDTH,GDMAXWIDTH)

        HBWIDTH=MIN(HBWIDTH,HBMAXWIDTH)
        HCWIDTH=MIN(HCWIDTH,HCMAXWIDTH)
C        HDWIDTH=MIN(HDWIDTH,HDMAXWIDTH)
```

```
                    WRITE(2,*)"*********AVERAGE    PLUS    TWO    STANDARD
DEVIATIONS*********"
      WRITE(2,*)"CATEG.     ","COUNT          ","CAPACITY     ",
     1          "LENGTH          ","WIDTH"

      WRITE(2,*)"AA",AACOUNT,AACAPACITY,AALENGTH,AAWIDTH
      WRITE(2,*)"AB",ABCOUNT,ABCAPACITY,ABLENGTH,ABWIDTH
      WRITE(2,*)"AC",ACCOUNT,ACCAPACITY,ACLENGTH,ACWIDTH
      WRITE(2,*)"AD",ADCOUNT,ADCAPACITY,ADLENGTH,ADWIDTH

      WRITE(2,*)"BA",BACOUNT,BACAPACITY,BALENGTH,BAWIDTH
      WRITE(2,*)"BB",BBCOUNT,BBCAPACITY,BBLENGTH,BBWIDTH
      WRITE(2,*)"BC",BCCOUNT,BCCAPACITY,BCLENGTH,BCWIDTH
      WRITE(2,*)"BD",BDCOUNT,BDCAPACITY,BDLENGTH,BDWIDTH

      WRITE(2,*)"CB",CBCOUNT,CBCAPACITY,CBLENGTH,CBWIDTH
      WRITE(2,*)"CC",CCCOUNT,CCCAPACITY,CCLENGTH,CCWIDTH
      WRITE(2,*)"CD",CDCOUNT,CDCAPACITY,CDLENGTH,CDWIDTH

      WRITE(2,*)"DB",DBCOUNT,DBCAPACITY,DBLENGTH,DBWIDTH
      WRITE(2,*)"DC",DCCOUNT,DCCAPACITY,DCLENGTH,DCWIDTH
      WRITE(2,*)"DD",DDCOUNT,DDCAPACITY,DDLENGTH,DDWIDTH

      WRITE(2,*)"EA",EACOUNT,EACAPACITY,EALENGTH,EAWIDTH
      WRITE(2,*)"EB",EBCOUNT,EBCAPACITY,EBLENGTH,EBWIDTH
      WRITE(2,*)"EC",ECCOUNT,ECCAPACITY,ECLENGTH,ECWIDTH
      WRITE(2,*)"ED",EDCOUNT,EDCAPACITY,EDLENGTH,EDWIDTH

      WRITE(2,*)"FC",FCCOUNT,FCCAPACITY,FCLENGTH,FCWIDTH
      WRITE(2,*)"FD",FDCOUNT,FDCAPACITY,FDLENGTH,FDWIDTH

      WRITE(2,*)"GC",GCCOUNT,GCCAPACITY,GCLENGTH,GCWIDTH
      WRITE(2,*)"GD",GDCOUNT,GDCAPACITY,GDLENGTH,GDWIDTH

      WRITE(2,*)"HB",HBCOUNT,HBCAPACITY,HBLENGTH,HBWIDTH
      WRITE(2,*)"HC",HCCOUNT,HCCAPACITY,HCLENGTH,HCWIDTH
      WRITE(2,*)"HD",HDCOUNT,HDCAPACITY,HDLENGTH,HDWIDTH


      END
```

Appendix II:

Barge Capacity Results

**********CATEGORY AVERAGE VALUES**********

CATEG.	COUNT	CAPACITY	LENGTH	WIDTH
AA	335.000000	223.129400	71.887470	20.699710
AB	811.000000	567.839100	62.983400	30.952040
AC	60.000000	1926.766000	87.573330	40.275000
AD	1.000000	500.571400	98.000000	55.000000
BA	52.000000	630.556600	112.594200	23.567300
BB	1605.000000	666.538300	122.286200	29.468540
BC	981.000000	1583.328000	142.572900	42.995820
BD	24.000000	1810.479000	133.825000	55.375000
CB	1034.000000	1160.919000	175.051900	26.058050
CC	313.000000	1979.785000	180.180800	42.700960
CD	1.000000	420.652200	188.000000	60.000000
DB	491.000000	1361.573000	195.002000	26.016290
DC	13075.000000	1844.535000	195.008700	35.100230
DD	1.000000	2642.667000	196.100000	54.100000
EA	3.000000	1155.556000	215.000000	25.000000
EB	3.000000	1257.392000	200.000000	26.000000
EC	6841.000000	2075.766000	202.292600	35.983460
ED	45.000000	5100.611000	242.677800	71.226670
FC	353.000000	3643.621000	265.482700	51.433440
FD	6.000000	3666.945000	268.350000	56.183330
GC	622.000000	4306.824000	295.333000	53.167680
GD	26.000000	4875.167000	297.296200	54.330760
HB	1.000000	82.500000	913.000000	26.000000
HC	35.000000	4837.305000	333.020000	52.439990
HD	2.000000	5504.164000	340.050000	54.550000

**********AVERAGE PLUS TWO STANDARD DEVIATIONS**********

CATEG.	COUNT	CAPACITY	LENGTH	WIDTH
AA	335.000000	608.962200	99.500000	25.700000
AB	811.000000	952.719300	75.870190	33.131920
AC	60.000000	4485.833000	99.400000	54.000000
AD	1.000000	500.571400	98.000000	55.000000
BA	52.000000	1419.585000	144.060200	25.000000
BB	1605.000000	1226.327000	150.860300	33.587460
BC	981.000000	3414.109000	174.000000	54.000000
BD	24.000000	3663.902000	160.000000	59.303160
CB	1034.000000	1868.193000	176.114700	26.975800
CC	313.000000	3652.683000	191.185600	54.000000
CD	1.000000	420.652200	188.000000	60.000000
DB	491.000000	1890.019000	195.092300	26.738360
DC	13075.000000	2715.042000	195.246100	37.486560
DD	1.000000	2642.667000	196.100000	54.100000
EA	3.000000	1155.556000	215.000000	25.000000
EB	3.000000	1375.000000	200.000000	26.000000
EC	6841.000000	3046.373000	221.033900	43.228270
ED	45.000000	7714.286000	250.000000	72.000000
FC	353.000000	5315.077000	279.257700	54.000000
FD	6.000000	4260.870000	285.000000	62.690450
GC	622.000000	6476.832000	300.000000	54.000000
GD	26.000000	7497.494000	297.900000	56.643690
HB	1.000000	82.500000	913.000000	26.000000
HC	35.000000	8382.554000	404.271700	54.000000
HD	2.000000	6349.505000	360.100000	55.822790

Appendix III:

Computer Program for Assigning the Number of Barges to Flotilla Columns and Rows

```
DECLARE SUB CATEGORY ()
DECLARE SUB CLASSIFY ()
DECLARE SUB CLASSIFY2 ()
DECLARE SUB DEVIATION ()
DECLARE SUB DIVBY1 ()
DECLARE SUB DIVBY2 ()
DECLARE SUB DIVBY3 ()
DECLARE SUB DIVBY4 ()
DECLARE SUB DIVBY5 ()
DECLARE SUB DIVBY6 ()
DECLARE SUB DIVBY7 ()
DECLARE SUB LABEL ()
DECLARE SUB MESSAGE ()
DECLARE SUB RESULTS ()
REM*************************************************************************
REM*
REM*                              AVGBARGE.BAS
REM*
REM*          Program to calculate the average number of barges
REM*           per flotilla column and row in each of 32 barge
REM*            dimension categories as specified by the U.S.
REM*                        Army Corps of Engineers
REM*
REM*                    Written by: Jeff Griffin, June 1994
REM*
REM*************************************************************************
REM
   COMMON SHARED FILEI$, FILEO$, RIVER$, MILE
   COMMON SHARED H%, I%, J%, K%, L%, M%, N%, O%
   COMMON SHARED FL(), FW(), NOC(), NB(), NBCL(), NBRW()
   COMMON SHARED FWAVG(), FLAVG(), NUM
   COMMON SHARED SUMNOC(), SUMNBRWxNOC(), SUMNBCLxNOC()
   COMMON SHARED CATAVGC(), CATAVGR(), MAXC(), MAXR()
   COMMON SHARED DEVNUMR(), STDR(), STDR2(), NUMBR
   COMMON SHARED DEVNUMC(), STDC(), STDC2()
   DIM FL(3000), FW(3000), NOC(3000), NB(3000), NBCL(3000), NBRW(3000)
   DIM FWAVG(3000), FLAVG(3000)
   DIM SUMNOC(32), SUMNBRWxNOC(32), SUMNBCLxNOC(32)
   DIM CATAVGC(32), CATAVGR(32), MAXC(32), MAXR(32)
   DIM DEVNUMR(32), STDR(32), STDR2(32)
   DIM DEVNUMC(32), STDC(32), STDC2(32)
REM
   CLS
   SCREEN 12
REM
REM*************************************************************************
REM*
*
REM*                INTIALIZE WEIGHTED AVERAGE SUM VARIABLES
*
REM*
*
REM*************************************************************************
REM
   FOR H% = 1 TO 32
      SUMNOC(H%) = 0
      SUMNBRWxNOC(H%) = 0
      SUMNBCLxNOC(H%) = 0
      MAXC(H%) = 0
      MAXR(H%) = 0
   NEXT H%
REM
REM*************************************************************************
REM*
*
```

```
REM*                    FILE INPUT/OUTPUT SPECIFICATION
*
REM*
*
REM*************************************************************************
REM
   COLOR 10
    LOCATE 4, 15: PRINT "AVERAGE NUMBER OF BARGES IN FLOTILLA COLUMN
PROGRAM"
   COLOR 15
   LINE (70, 80)-(580, 200), 9, B
   LOCATE 7, 14: INPUT "What is the name of the file to be processed";
FILEI$
   OPEN FILEI$ FOR INPUT AS #1
    LOCATE 9, 14: INPUT "What is the name of the file to be output";
FILEO$
   OPEN FILEO$ FOR OUTPUT AS #2
    LOCATE 11, 14: INPUT "What is the river and marker designation";
RIVER$, MILE
REM
REM*************************************************************************
REM*
*
REM*                            FLOTILLA DATA INPUT
*
REM*
*
REM*************************************************************************
REM
   J% = 1
   INPUT #1, FL(1), FW(1), NOC(1), NB(1)
   DO WHILE FL(J%) > 0
      J% = J% + 1
      INPUT #1, FL(J%), FW(J%), NOC(J%), NB(J%)
   LOOP
   CALL MESSAGE
REM
REM*************************************************************************
REM*
*
REM*       MODIFICATION OF NUMBER OF BARGES IN FLOTILLA TO ADJUST
*
REM*            FOR NUMBER OF BARGES EXPECTED IN COLUMN AND ROW
*
REM*
*
REM*************************************************************************
REM
   K% = J% - 1
   I% = 1
   DO WHILE I% <= K%
REM
REM   CHECK FLOTILLA WIDTH TO BE WITHIN A REASONABLE RANGE
REM
      IF FW(I%) >= 20 AND FW(I%) <= 245 THEN
REM
REM      CHECK FLOTILLA WIDTH TO BE LESS THAN 50 FEET
REM
         IF FW(I%) < 50 THEN
            CALL DIVBY1
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE 50 FEET
REM
         IF FW(I%) = 50 THEN
```

```
            IF NB(I%) > 5 THEN
               CALL DIVBY2
            ELSE
               CALL DIVBY1
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 50 AND 60 FEET, EXCLUSIVE
REM
         IF FW(I%) > 50 AND FW(I%) < 60 THEN
            IF FL(I%) < 100 AND NB(I%) > 1 THEN
               CALL DIVBY2
            ELSE
               CALL DIVBY1
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE 60 FEET
REM
         IF FW(I%) = 60 THEN
            IF FL(I%) > 500 AND NB(I%) <= 3 THEN
               CALL DIVBY1
            ELSE
               CALL DIVBY2
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 60 AND 70 FEET, EXCLUSIVE
REM
         IF FW(I%) > 60 AND FW(I%) < 70 THEN
            IF FL(I%) >= 600 AND NB(I%) <= 5 THEN
               CALL DIVBY1
            ELSE
               CALL DIVBY2
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE 70 FEET
REM
         IF FW(I%) = 70 THEN
            CALL DIVBY2
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 70 AND 85 FEET, EXCLUSIVE
REM
         IF FW(I%) > 70 AND FW(I%) < 85 THEN
            IF NB(I%) >= 4 THEN
               CALL DIVBY2
            ELSE
               CALL DIVBY1
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 85 AND EXCLUSIVE 105 FEET
REM
         IF FW(I%) >= 85 AND FW(I%) < 105 THEN
            CALL DIVBY2
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE 105 FEET
REM
         IF FW(I%) = 105 THEN
            IF NB(I%) <= 6 THEN
               CALL DIVBY2
            ELSE
```

```
            CALL DIVBY3
         END IF
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN EXCLUSIVE 105 AND 110 FEET
REM
      IF FW(I%) > 105 AND FW(I%) <= 110 THEN
         CALL DIVBY2
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 110 AND 120 FEET, EXCLUSIVE
REM
      IF FW(I%) > 110 AND FW(I%) < 120 THEN
         IF NB(I%) > 21 THEN
            CALL DIVBY3
         ELSE
            CALL DIVBY2
         END IF
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 120 AND 122 FEET,INCLUSIVE
REM
      IF FW(I%) >= 120 AND FW(I%) <= 122 THEN
         CALL DIVBY4
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 122 AND 140 FEET, EXCLUSIVE
REM
      IF FW(I%) > 122 AND FW(I%) < 140 THEN
         CALL DIVBY3
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE 140 FEET
REM
      IF FW(I%) = 140 THEN
         CALL DIVBY4
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 140 AND INCLUSIVE 165 FEET
REM
      IF FW(I%) > 140 AND FW(I%) <= 165 THEN
         CALL DIVBY3
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 165 AND 175 FEET, EXCLUSIVE
REM
      IF FW(I%) > 165 AND FW(I%) < 175 THEN
         CALL DIVBY4
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE 175 FEET
REM
      IF FW(I%) = 175 THEN
         IF NB(I%) <= 10 THEN
            CALL DIVBY4
         ELSE
            CALL DIVBY5
         END IF
      END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 175 AND INCLUSIVE 180 FEET
REM
      IF FW(I%) > 175 AND FW(I%) <= 180 THEN
         IF NB(I%) <= 10 THEN
            CALL DIVBY2
```

```
            END IF
            IF NB(I%) > 10 AND NB(I%) < 19 THEN
               CALL DIVBY4
            END IF
            IF NB(I%) >= 19 THEN
               CALL DIVBY5
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 180 AND 200 FEET, EXCLUSIVE
REM
         IF FW(I%) > 180 AND FW(I%) < 200 THEN
            IF NB(I%) < 11 THEN
               CALL DIVBY4
            ELSE
               CALL DIVBY5
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN INCLUSIVE 200 AND 233 FEET
REM
         IF FW(I%) >= 200 AND FW(I%) < 233 THEN
            IF FW(I%) = 210 AND NB(I%) > 20 THEN
               CALL DIVBY6
            ELSE
               CALL DIVBY4
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 233 AND 290 FEET INCLUSIVE
REM
         IF FW(I%) >= 233 AND FW(I%) <= 290 THEN
            IF FW(I%) = 245 AND NB(I%) > 24 THEN
               CALL DIVBY7
            ELSE
               CALL DIVBY5
            END IF
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE BETWEEN 290 AND INCLUSIVE 330 FEET
REM
         IF FW(I%) > 290 AND FW(I%) <= 330 THEN
            CALL DIVBY6
         END IF
REM
REM      CHECK FLOTILLA WIDTH TO BE GREATER THAN 330 FEET
REM
         IF FW(I%) > 330 THEN
            CALL DIVBY7
         END IF
REM
         IF NBCL(I%) < 1 THEN NBCL(I%) = 1
REM
REM*************************************************************************
REM*
*
REM*     CALCULATE THE AVERAGE BARGE LENGTH IN THE FLOTILLA COLUMN
*
REM*
*
REM*************************************************************************
REM
         FLAVG(I%) = FL(I%) / NBCL(I%)
REM
REM*************************************************************************
```

```
REM*
*
REM*     DETERMINE THE APPROPRIATE BARGE CATEGORY FOR THE FLOTILLA
*
REM*       BASED UPON THE AVERAGE BARGE LENGTH CALCULATED ABOVE
*
REM*
*
REM*************************************************************************
REM
         CALL CLASSIFY
REM
REM*************************************************************************
REM*
*
REM*          SUM THE NUMBER OF BARGES PER ROW AND THE NUMBER OF
*
REM*           OCCURRENCES BASED ON THE CATEGORY ASSIGNED ABOVE
*
REM*
*
REM*************************************************************************
REM
         CALL CATEGORY
REM
      END IF
   I% = I% + 1
   LOOP
REM
REM*************************************************************************
REM*
*
REM*                      CALCULATE THE WEIGHTED AVERAGE
*
REM*
*
REM*************************************************************************
REM
   FOR L% = 1 TO 32
      IF SUMNBRWxNOC(L%) = 0 THEN
         CATAVGR(L%) = 0
      ELSE
         CATAVGR(L%) = SUMNBRWxNOC(L%) / SUMNOC(L%)
      END IF
      IF SUMNBCLxNOC(L%) = 0 THEN
         CATAVGC(L%) = 0
      ELSE
         CATAVGC(L%) = SUMNBCLxNOC(L%) / SUMNOC(L%)
      END IF
   NEXT L%
REM
REM*************************************************************************
REM*
*
REM*             CALCULATE THE STANDARD DEVIATION DENOMINATOR
*
REM*
*
REM*************************************************************************
REM
   FOR N% = 1 TO K%
      CALL CLASSIFY2
      CALL DEVIATION
   NEXT N%
REM
```

```
REM*************************************************************************
REM*
*
REM*   CALCULATE THE STANDARD DEVIATION FOR EACH OF THE 32 CATEGORIES
*
REM*
*
REM*************************************************************************
REM
   FOR O% = 1 TO 32
      IF SUMNOC(O%) <= 1 THEN
         STDR(O%) = 0
         STDC(O%) = 0
      ELSE
         STDR(O%) = (DEVNUMR(O%) / (SUMNOC(O%) - 1)) ^ .5
         STDC(O%) = (DEVNUMC(O%) / (SUMNOC(O%) - 1)) ^ .5
      END IF
      STDR2(O%) = 2 * STDR(O%)
      STDC2(O%) = 2 * STDC(O%)
   NEXT O%
REM
REM*************************************************************************
REM*
*
REM*              PRINT THE RESULTS TO THE SPECIFIED FILE
*
REM*
*
REM*************************************************************************
REM
   CALL RESULTS
REM
   END
   STOP

SUB CATEGORY
   SELECT CASE NUM
      CASE 1
         SUMNOC(1) = SUMNOC(1) + NOC(I%)
         SUMNBRWxNOC(1) = SUMNBRWxNOC(1) + (NBRW(I%) * NOC(I%))
         SUMNBCLxNOC(1) = SUMNBCLxNOC(1) + (NBCL(I%) * NOC(I%))
         IF NBCL(I%) > MAXC(1) THEN
            MAXC(1) = NBCL(I%)
         ELSE
            MAXC(1) = MAXC(1)
         END IF
         IF NBRW(I%) > MAXR(1) THEN
            MAXR(1) = NBRW(I%)
         ELSE
            MAXR(1) = MAXR(1)
         END IF
      CASE 2
         SUMNOC(2) = SUMNOC(2) + NOC(I%)
         SUMNBRWxNOC(2) = SUMNBRWxNOC(2) + (NBRW(I%) * NOC(I%))
         SUMNBCLxNOC(2) = SUMNBCLxNOC(2) + (NBCL(I%) * NOC(I%))
         IF NBCL(I%) > MAXC(2) THEN
            MAXC(2) = NBCL(I%)
         ELSE
            MAXC(2) = MAXC(2)
         END IF
         IF NBRW(I%) > MAXR(2) THEN
            MAXR(2) = NBRW(I%)
         ELSE
            MAXR(2) = MAXR(2)
         END IF
```

```
CASE 3
   SUMNOC(3) = SUMNOC(3) + NOC(I%)
   SUMNBRWxNOC(3) = SUMNBRWxNOC(3) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(3) = SUMNBCLxNOC(3) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(3) THEN
      MAXC(3) = NBCL(I%)
   ELSE
      MAXC(3) = MAXC(3)
   END IF
   IF NBRW(I%) > MAXR(3) THEN
      MAXR(3) = NBRW(I%)
   ELSE
      MAXR(3) = MAXR(3)
   END IF
CASE 4
   SUMNOC(4) = SUMNOC(4) + NOC(I%)
   SUMNBRWxNOC(4) = SUMNBRWxNOC(4) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(4) = SUMNBCLxNOC(4) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(4) THEN
      MAXC(4) = NBCL(I%)
   ELSE
      MAXC(4) = MAXC(4)
   END IF
   IF NBRW(I%) > MAXR(4) THEN
      MAXR(4) = NBRW(I%)
   ELSE
      MAXR(4) = MAXR(4)
   END IF
CASE 5
   SUMNOC(5) = SUMNOC(5) + NOC(I%)
   SUMNBRWxNOC(5) = SUMNBRWxNOC(5) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(5) = SUMNBCLxNOC(5) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(5) THEN
      MAXC(5) = NBCL(I%)
   ELSE
      MAXC(5) = MAXC(5)
   END IF
   IF NBRW(I%) > MAXR(5) THEN
      MAXR(5) = NBRW(I%)
   ELSE
      MAXR(5) = MAXR(5)
   END IF
CASE 6
   SUMNOC(6) = SUMNOC(6) + NOC(I%)
   SUMNBRWxNOC(6) = SUMNBRWxNOC(6) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(6) = SUMNBCLxNOC(6) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(6) THEN
      MAXC(6) = NBCL(I%)
   ELSE
      MAXC(6) = MAXC(6)
   END IF
   IF NBRW(I%) > MAXR(6) THEN
      MAXR(6) = NBRW(I%)
   ELSE
      MAXR(6) = MAXR(6)
   END IF
CASE 7
   SUMNOC(7) = SUMNOC(7) + NOC(I%)
   SUMNBRWxNOC(7) = SUMNBRWxNOC(7) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(7) = SUMNBCLxNOC(7) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(7) THEN
      MAXC(7) = NBCL(I%)
   ELSE
      MAXC(7) = MAXC(7)
   END IF
```

```
      IF NBRW(I%) > MAXR(7) THEN
         MAXR(7) = NBRW(I%)
      ELSE
         MAXR(7) = MAXR(7)
      END IF
   CASE 8
      SUMNOC(8) = SUMNOC(8) + NOC(I%)
      SUMNBRWxNOC(8) = SUMNBRWxNOC(8) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(8) = SUMNBCLxNOC(8) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(8) THEN
         MAXC(8) = NBCL(I%)
      ELSE
         MAXC(8) = MAXC(8)
      END IF
      IF NBRW(I%) > MAXR(8) THEN
         MAXR(8) = NBRW(I%)
      ELSE
         MAXR(8) = MAXR(8)
      END IF
   CASE 9
      SUMNOC(9) = SUMNOC(9) + NOC(I%)
      SUMNBRWxNOC(9) = SUMNBRWxNOC(9) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(9) = SUMNBCLxNOC(9) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(9) THEN
         MAXC(9) = NBCL(I%)
      ELSE
         MAXC(9) = MAXC(9)
      END IF
      IF NBRW(I%) > MAXR(9) THEN
         MAXR(9) = NBRW(I%)
      ELSE
         MAXR(9) = MAXR(9)
      END IF
   CASE 10
      SUMNOC(10) = SUMNOC(10) + NOC(I%)
      SUMNBRWxNOC(10) = SUMNBRWxNOC(10) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(10) = SUMNBCLxNOC(10) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(10) THEN
         MAXC(10) = NBCL(I%)
      ELSE
         MAXC(10) = MAXC(10)
      END IF
      IF NBRW(I%) > MAXR(10) THEN
         MAXR(10) = NBRW(I%)
      ELSE
         MAXR(10) = MAXR(10)
      END IF
   CASE 11
      SUMNOC(11) = SUMNOC(11) + NOC(I%)
      SUMNBRWxNOC(11) = SUMNBRWxNOC(11) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(11) = SUMNBCLxNOC(11) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(11) THEN
         MAXC(11) = NBCL(I%)
      ELSE
         MAXC(11) = MAXC(11)
      END IF
      IF NBRW(I%) > MAXR(11) THEN
         MAXR(11) = NBRW(I%)
      ELSE
         MAXR(11) = MAXR(11)
      END IF
   CASE 12
      SUMNOC(12) = SUMNOC(12) + NOC(I%)
      SUMNBRWxNOC(12) = SUMNBRWxNOC(12) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(12) = SUMNBCLxNOC(12) + (NBCL(I%) * NOC(I%))
```

```
      IF NBCL(I%) > MAXC(12) THEN
         MAXC(12) = NBCL(I%)
      ELSE
         MAXC(12) = MAXC(12)
      END IF
      IF NBRW(I%) > MAXR(12) THEN
         MAXR(12) = NBRW(I%)
      ELSE
         MAXR(12) = MAXR(12)
      END IF
   CASE 13
      SUMNOC(13) = SUMNOC(13) + NOC(I%)
      SUMNBRWxNOC(13) = SUMNBRWxNOC(13) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(13) = SUMNBCLxNOC(13) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(13) THEN
         MAXC(13) = NBCL(I%)
      ELSE
         MAXC(13) = MAXC(13)
      END IF
      IF NBRW(I%) > MAXR(13) THEN
         MAXR(13) = NBRW(I%)
      ELSE
         MAXR(13) = MAXR(13)
      END IF
   CASE 14
      SUMNOC(14) = SUMNOC(14) + NOC(I%)
      SUMNBRWxNOC(14) = SUMNBRWxNOC(14) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(14) = SUMNBCLxNOC(14) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(14) THEN
         MAXC(14) = NBCL(I%)
      ELSE
         MAXC(14) = MAXC(14)
      END IF
      IF NBRW(I%) > MAXR(14) THEN
         MAXR(14) = NBRW(I%)
      ELSE
         MAXR(14) = MAXR(14)
      END IF
   CASE 15
      SUMNOC(15) = SUMNOC(15) + NOC(I%)
      SUMNBRWxNOC(15) = SUMNBRWxNOC(15) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(15) = SUMNBCLxNOC(15) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(15) THEN
         MAXC(15) = NBCL(I%)
      ELSE
         MAXC(15) = MAXC(15)
      END IF
      IF NBRW(I%) > MAXR(15) THEN
         MAXR(15) = NBRW(I%)
      ELSE
         MAXR(15) = MAXR(15)
      END IF
   CASE 16
      SUMNOC(16) = SUMNOC(16) + NOC(I%)
      SUMNBRWxNOC(16) = SUMNBRWxNOC(16) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(16) = SUMNBCLxNOC(16) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(16) THEN
         MAXC(16) = NBCL(I%)
      ELSE
         MAXC(16) = MAXC(16)
      END IF
      IF NBRW(I%) > MAXR(16) THEN
         MAXR(16) = NBRW(I%)
      ELSE
         MAXR(16) = MAXR(16)
```

```
   END IF
CASE 17
   SUMNOC(17) = SUMNOC(17) + NOC(I%)
   SUMNBRWxNOC(17) = SUMNBRWxNOC(17) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(17) = SUMNBCLxNOC(17) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(17) THEN
      MAXC(17) = NBCL(I%)
   ELSE
      MAXC(17) = MAXC(17)
   END IF
   IF NBRW(I%) > MAXR(17) THEN
      MAXR(17) = NBRW(I%)
   ELSE
      MAXR(17) = MAXR(17)
   END IF
CASE 18
   SUMNOC(18) = SUMNOC(18) + NOC(I%)
   SUMNBRWxNOC(18) = SUMNBRWxNOC(18) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(18) = SUMNBCLxNOC(18) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(18) THEN
      MAXC(18) = NBCL(I%)
   ELSE
      MAXC(18) = MAXC(18)
   END IF
   IF NBRW(I%) > MAXR(18) THEN
      MAXR(18) = NBRW(I%)
   ELSE
      MAXR(18) = MAXR(18)
   END IF
CASE 19
   SUMNOC(19) = SUMNOC(19) + NOC(I%)
   SUMNBRWxNOC(19) = SUMNBRWxNOC(19) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(19) = SUMNBCLxNOC(19) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(19) THEN
      MAXC(19) = NBCL(I%)
   ELSE
      MAXC(19) = MAXC(19)
   END IF
   IF NBRW(I%) > MAXR(19) THEN
      MAXR(19) = NBRW(I%)
   ELSE
      MAXR(19) = MAXR(19)
   END IF
CASE 20
   SUMNOC(20) = SUMNOC(20) + NOC(I%)
   SUMNBRWxNOC(20) = SUMNBRWxNOC(20) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(20) = SUMNBCLxNOC(20) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(20) THEN
      MAXC(20) = NBCL(I%)
   ELSE
      MAXC(20) = MAXC(20)
   END IF
   IF NBRW(I%) > MAXR(20) THEN
      MAXR(20) = NBRW(I%)
   ELSE
      MAXR(20) = MAXR(20)
   END IF
CASE 21
   SUMNOC(21) = SUMNOC(21) + NOC(I%)
   SUMNBRWxNOC(21) = SUMNBRWxNOC(21) + (NBRW(I%) * NOC(I%))
   SUMNBCLxNOC(21) = SUMNBCLxNOC(21) + (NBCL(I%) * NOC(I%))
   IF NBCL(I%) > MAXC(21) THEN
      MAXC(21) = NBCL(I%)
   ELSE
      MAXC(21) = MAXC(21)
```

```
      END IF
      IF NBRW(I%) > MAXR(21) THEN
         MAXR(21) = NBRW(I%)
      ELSE
         MAXR(21) = MAXR(21)
      END IF
   CASE 22
      SUMNOC(22) = SUMNOC(22) + NOC(I%)
      SUMNBRWxNOC(22) = SUMNBRWxNOC(22) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(22) = SUMNBCLxNOC(22) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(22) THEN
         MAXC(22) = NBCL(I%)
      ELSE
         MAXC(22) = MAXC(22)
      END IF
      IF NBRW(I%) > MAXR(22) THEN
         MAXR(22) = NBRW(I%)
      ELSE
         MAXR(22) = MAXR(22)
      END IF
   CASE 23
      SUMNOC(23) = SUMNOC(23) + NOC(I%)
      SUMNBRWxNOC(23) = SUMNBRWxNOC(23) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(23) = SUMNBCLxNOC(23) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(23) THEN
         MAXC(23) = NBCL(I%)
      ELSE
         MAXC(23) = MAXC(23)
      END IF
      IF NBRW(I%) > MAXR(23) THEN
         MAXR(23) = NBRW(I%)
      ELSE
         MAXR(23) = MAXR(23)
      END IF
   CASE 24
      SUMNOC(24) = SUMNOC(24) + NOC(I%)
      SUMNBRWxNOC(24) = SUMNBRWxNOC(24) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(24) = SUMNBCLxNOC(24) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(24) THEN
         MAXC(24) = NBCL(I%)
      ELSE
         MAXC(24) = MAXC(24)
      END IF
      IF NBRW(I%) > MAXR(24) THEN
         MAXR(24) = NBRW(I%)
      ELSE
         MAXR(24) = MAXR(24)
      END IF
   CASE 25
      SUMNOC(25) = SUMNOC(25) + NOC(I%)
      SUMNBRWxNOC(25) = SUMNBRWxNOC(25) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(25) = SUMNBCLxNOC(25) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(25) THEN
         MAXC(25) = NBCL(I%)
      ELSE
         MAXC(25) = MAXC(25)
      END IF
      IF NBRW(I%) > MAXR(25) THEN
         MAXR(25) = NBRW(I%)
      ELSE
         MAXR(25) = MAXR(25)
      END IF
   CASE 26
      SUMNOC(26) = SUMNOC(26) + NOC(I%)
      SUMNBRWxNOC(26) = SUMNBRWxNOC(26) + (NBRW(I%) * NOC(I%))
```

```
      SUMNBCLxNOC(26) = SUMNBCLxNOC(26) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(26) THEN
         MAXC(26) = NBCL(I%)
      ELSE
         MAXC(26) = MAXC(26)
      END IF
      IF NBRW(I%) > MAXR(26) THEN
         MAXR(26) = NBRW(I%)
      ELSE
         MAXR(26) = MAXR(26)
      END IF
   CASE 27
      SUMNOC(27) = SUMNOC(27) + NOC(I%)
      SUMNBRWxNOC(27) = SUMNBRWxNOC(27) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(27) = SUMNBCLxNOC(27) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(27) THEN
         MAXC(27) = NBCL(I%)
      ELSE
         MAXC(27) = MAXC(27)
      END IF
      IF NBRW(I%) > MAXR(27) THEN
         MAXR(27) = NBRW(I%)
      ELSE
         MAXR(27) = MAXR(27)
      END IF
   CASE 28
      SUMNOC(28) = SUMNOC(28) + NOC(I%)
      SUMNBRWxNOC(28) = SUMNBRWxNOC(28) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(28) = SUMNBCLxNOC(28) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(28) THEN
         MAXC(28) = NBCL(I%)
      ELSE
         MAXC(28) = MAXC(28)
      END IF
      IF NBRW(I%) > MAXR(28) THEN
         MAXR(28) = NBRW(I%)
      ELSE
         MAXR(28) = MAXR(28)
      END IF
   CASE 29
      SUMNOC(29) = SUMNOC(29) + NOC(I%)
      SUMNBRWxNOC(29) = SUMNBRWxNOC(29) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(29) = SUMNBCLxNOC(29) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(29) THEN
         MAXC(29) = NBCL(I%)
      ELSE
         MAXC(29) = MAXC(29)
      END IF
      IF NBRW(I%) > MAXR(29) THEN
         MAXR(29) = NBRW(I%)
      ELSE
         MAXR(29) = MAXR(29)
      END IF
   CASE 30
      SUMNOC(30) = SUMNOC(30) + NOC(I%)
      SUMNBRWxNOC(30) = SUMNBRWxNOC(30) + (NBRW(I%) * NOC(I%))
      SUMNBCLxNOC(30) = SUMNBCLxNOC(30) + (NBCL(I%) * NOC(I%))
      IF NBCL(I%) > MAXC(30) THEN
         MAXC(30) = NBCL(I%)
      ELSE
         MAXC(30) = MAXC(30)
      END IF
      IF NBRW(I%) > MAXR(30) THEN
         MAXR(30) = NBRW(I%)
      ELSE
```

```
            MAXR(30) = MAXR(30)
         END IF
      CASE 31
         SUMNOC(31) = SUMNOC(31) + NOC(I%)
         SUMNBRWxNOC(31) = SUMNBRWxNOC(31) + (NBRW(I%) * NOC(I%))
         SUMNBCLxNOC(31) = SUMNBCLxNOC(31) + (NBCL(I%) * NOC(I%))
         IF NBCL(I%) > MAXC(31) THEN
            MAXC(31) = NBCL(I%)
         ELSE
            MAXC(31) = MAXC(31)
         END IF
         IF NBRW(I%) > MAXR(31) THEN
            MAXR(31) = NBRW(I%)
         ELSE
            MAXR(31) = MAXR(31)
         END IF
      CASE 32
         SUMNOC(32) = SUMNOC(32) + NOC(I%)
         SUMNBRWxNOC(32) = SUMNBRWxNOC(32) + (NBRW(I%) * NOC(I%))
         SUMNBCLxNOC(32) = SUMNBCLxNOC(32) + (NBCL(I%) * NOC(I%))
         IF NBCL(I%) > MAXC(32) THEN
            MAXC(32) = NBCL(I%)
         ELSE
            MAXC(32) = MAXC(32)
         END IF
         IF NBRW(I%) > MAXR(32) THEN
            MAXR(32) = NBRW(I%)
         ELSE
            MAXR(32) = MAXR(32)
         END IF
   END SELECT
END SUB

SUB CLASSIFY
   IF FLAVG(I%) < 100 THEN
      IF FWAVG(I%) < 26 THEN NUM = 1
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 2
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 3
      IF FWAVG(I%) > 54 THEN NUM = 4
   END IF
   IF FLAVG(I%) >= 100 AND FLAVG(I%) <= 174 THEN
      IF FWAVG(I%) < 26 THEN NUM = 5
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 6
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 7
      IF FWAVG(I%) > 54 THEN NUM = 8
   END IF
   IF FLAVG(I%) > 174 AND FLAVG(I%) <= 194 THEN
      IF FWAVG(I%) < 26 THEN NUM = 9
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 10
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 11
      IF FWAVG(I%) > 54 THEN NUM = 12
   END IF
   IF FLAVG(I%) > 194 AND FLAVG(I%) <= 199 THEN
      IF FWAVG(I%) < 26 THEN NUM = 13
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 14
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 15
      IF FWAVG(I%) > 54 THEN NUM = 16
   END IF
   IF FLAVG(I%) > 199 AND FLAVG(I%) <= 259 THEN
      IF FWAVG(I%) < 26 THEN NUM = 17
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 18
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 19
      IF FWAVG(I%) > 54 THEN NUM = 20
   END IF
   IF FLAVG(I%) > 259 AND FLAVG(I%) <= 289 THEN
```

```
      IF FWAVG(I%) < 26 THEN NUM = 21
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 22
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 23
      IF FWAVG(I%) > 54 THEN NUM = 24
   END IF
   IF FLAVG(I%) > 289 AND FLAVG(I%) <= 300 THEN
      IF FWAVG(I%) < 26 THEN NUM = 25
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 26
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 27
      IF FWAVG(I%) > 54 THEN NUM = 28
   END IF
   IF FLAVG(I%) > 300 THEN
      IF FWAVG(I%) < 26 THEN NUM = 29
      IF FWAVG(I%) >= 26 AND FWAVG(I%) < 35 THEN NUM = 30
      IF FWAVG(I%) >= 35 AND FWAVG(I%) <= 54 THEN NUM = 31
      IF FWAVG(I%) > 54 THEN NUM = 32
   END IF
END SUB

SUB CLASSIFY2
   IF FLAVG(N%) < 100 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 1
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 2
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 3
      IF FWAVG(N%) > 54 THEN NUMBR = 4
   END IF
   IF FLAVG(N%) >= 100 AND FLAVG(N%) <= 174 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 5
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 6
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 7
      IF FWAVG(N%) > 54 THEN NUMBR = 8
   END IF
   IF FLAVG(N%) > 174 AND FLAVG(N%) <= 194 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 9
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 10
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 11
      IF FWAVG(N%) > 54 THEN NUMBR = 12
   END IF
   IF FLAVG(N%) > 194 AND FLAVG(N%) <= 199 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 13
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 14
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 15
      IF FWAVG(N%) > 54 THEN NUMBR = 16
   END IF
   IF FLAVG(N%) > 199 AND FLAVG(N%) <= 259 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 17
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 18
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 19
      IF FWAVG(N%) > 54 THEN NUMBR = 20
   END IF
   IF FLAVG(N%) > 259 AND FLAVG(N%) <= 289 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 21
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 22
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 23
      IF FWAVG(N%) > 54 THEN NUMBR = 24
   END IF
   IF FLAVG(N%) > 289 AND FLAVG(N%) <= 300 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 25
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 26
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 27
      IF FWAVG(N%) > 54 THEN NUMBR = 28
   END IF
   IF FLAVG(N%) > 300 THEN
      IF FWAVG(N%) < 26 THEN NUMBR = 29
      IF FWAVG(N%) >= 26 AND FWAVG(N%) < 35 THEN NUMBR = 30
```

```
      IF FWAVG(N%) >= 35 AND FWAVG(N%) <= 54 THEN NUMBR = 31
      IF FWAVG(N%) > 54 THEN NUMBR = 32
   END IF
END SUB

SUB DEVIATION
   SELECT CASE NUMBR
      CASE 1
         DEVNUMR(1) = DEVNUMR(1) + (NBRW(N%) - CATAVGR(1)) ^ 2
         DEVNUMC(1) = DEVNUMC(1) + (NBCL(N%) - CATAVGC(1)) ^ 2
      CASE 2
         DEVNUMR(2) = DEVNUMR(2) + (NBRW(N%) - CATAVGR(2)) ^ 2
         DEVNUMC(2) = DEVNUMC(2) + (NBCL(N%) - CATAVGC(2)) ^ 2
      CASE 3
         DEVNUMR(3) = DEVNUMR(3) + (NBRW(N%) - CATAVGR(3)) ^ 2
         DEVNUMC(3) = DEVNUMC(3) + (NBCL(N%) - CATAVGC(3)) ^ 2
      CASE 4
         DEVNUMR(4) = DEVNUMR(4) + (NBRW(N%) - CATAVGR(4)) ^ 2
         DEVNUMC(4) = DEVNUMC(4) + (NBCL(N%) - CATAVGC(4)) ^ 2
      CASE 5
         DEVNUMR(5) = DEVNUMR(5) + (NBRW(N%) - CATAVGR(5)) ^ 2
         DEVNUMC(5) = DEVNUMC(5) + (NBCL(N%) - CATAVGC(5)) ^ 2
      CASE 6
         DEVNUMR(6) = DEVNUMR(6) + (NBRW(N%) - CATAVGR(6)) ^ 2
         DEVNUMC(6) = DEVNUMC(6) + (NBCL(N%) - CATAVGC(6)) ^ 2
      CASE 7
         DEVNUMR(7) = DEVNUMR(7) + (NBRW(N%) - CATAVGR(7)) ^ 2
         DEVNUMC(7) = DEVNUMC(7) + (NBCL(N%) - CATAVGC(7)) ^ 2
      CASE 8
         DEVNUMR(8) = DEVNUMR(8) + (NBRW(N%) - CATAVGR(8)) ^ 2
         DEVNUMC(8) = DEVNUMC(8) + (NBCL(N%) - CATAVGC(8)) ^ 2
      CASE 9
         DEVNUMR(9) = DEVNUMR(9) + (NBRW(N%) - CATAVGR(9)) ^ 2
         DEVNUMC(9) = DEVNUMC(9) + (NBCL(N%) - CATAVGC(9)) ^ 2
      CASE 10
         DEVNUMR(10) = DEVNUMR(10) + (NBRW(N%) - CATAVGR(10)) ^ 2
         DEVNUMC(10) = DEVNUMC(10) + (NBCL(N%) - CATAVGC(10)) ^ 2
      CASE 11
         DEVNUMR(11) = DEVNUMR(11) + (NBRW(N%) - CATAVGR(11)) ^ 2
         DEVNUMC(11) = DEVNUMC(11) + (NBCL(N%) - CATAVGC(11)) ^ 2
      CASE 12
         DEVNUMR(12) = DEVNUMR(12) + (NBRW(N%) - CATAVGR(12)) ^ 2
         DEVNUMC(12) = DEVNUMC(12) + (NBCL(N%) - CATAVGC(12)) ^ 2
      CASE 13
         DEVNUMR(13) = DEVNUMR(13) + (NBRW(N%) - CATAVGR(13)) ^ 2
         DEVNUMC(13) = DEVNUMC(13) + (NBCL(N%) - CATAVGC(13)) ^ 2
      CASE 14
         DEVNUMR(14) = DEVNUMR(14) + (NBRW(N%) - CATAVGR(14)) ^ 2
         DEVNUMC(14) = DEVNUMC(14) + (NBCL(N%) - CATAVGC(14)) ^ 2
      CASE 15
         DEVNUMR(15) = DEVNUMR(15) + (NBRW(N%) - CATAVGR(15)) ^ 2
         DEVNUMC(15) = DEVNUMC(15) + (NBCL(N%) - CATAVGC(15)) ^ 2
      CASE 16
         DEVNUMR(16) = DEVNUMR(16) + (NBRW(N%) - CATAVGR(16)) ^ 2
         DEVNUMC(16) = DEVNUMC(16) + (NBCL(N%) - CATAVGC(16)) ^ 2
      CASE 17
         DEVNUMR(17) = DEVNUMR(17) + (NBRW(N%) - CATAVGR(17)) ^ 2
         DEVNUMC(17) = DEVNUMC(17) + (NBCL(N%) - CATAVGC(17)) ^ 2
      CASE 18
         DEVNUMR(18) = DEVNUMR(18) + (NBRW(N%) - CATAVGR(18)) ^ 2
         DEVNUMC(18) = DEVNUMC(18) + (NBCL(N%) - CATAVGC(18)) ^ 2
      CASE 19
         DEVNUMR(19) = DEVNUMR(19) + (NBRW(N%) - CATAVGR(19)) ^ 2
         DEVNUMC(19) = DEVNUMC(19) + (NBCL(N%) - CATAVGC(19)) ^ 2
      CASE 20
```

```
            DEVNUMR(20) = DEVNUMR(20) + (NBRW(N%) - CATAVGR(20)) ^ 2
            DEVNUMC(20) = DEVNUMC(20) + (NBCL(N%) - CATAVGC(20)) ^ 2
         CASE 21
            DEVNUMR(21) = DEVNUMR(21) + (NBRW(N%) - CATAVGR(21)) ^ 2
            DEVNUMC(21) = DEVNUMC(21) + (NBCL(N%) - CATAVGC(21)) ^ 2
         CASE 22
            DEVNUMR(22) = DEVNUMR(22) + (NBRW(N%) - CATAVGR(22)) ^ 2
            DEVNUMC(22) = DEVNUMC(22) + (NBCL(N%) - CATAVGC(22)) ^ 2
         CASE 23
            DEVNUMR(23) = DEVNUMR(23) + (NBRW(N%) - CATAVGR(23)) ^ 2
            DEVNUMC(23) = DEVNUMC(23) + (NBCL(N%) - CATAVGC(23)) ^ 2
         CASE 24
            DEVNUMR(24) = DEVNUMR(24) + (NBRW(N%) - CATAVGR(24)) ^ 2
            DEVNUMC(24) = DEVNUMC(24) + (NBCL(N%) - CATAVGC(24)) ^ 2
         CASE 25
            DEVNUMR(25) = DEVNUMR(25) + (NBRW(N%) - CATAVGR(25)) ^ 2
            DEVNUMC(25) = DEVNUMC(25) + (NBCL(N%) - CATAVGC(25)) ^ 2
         CASE 26
            DEVNUMR(26) = DEVNUMR(26) + (NBRW(N%) - CATAVGR(26)) ^ 2
            DEVNUMC(26) = DEVNUMC(26) + (NBCL(N%) - CATAVGC(26)) ^ 2
         CASE 27
            DEVNUMR(27) = DEVNUMR(27) + (NBRW(N%) - CATAVGR(27)) ^ 2
            DEVNUMC(27) = DEVNUMC(27) + (NBCL(N%) - CATAVGC(27)) ^ 2
         CASE 28
            DEVNUMR(28) = DEVNUMR(28) + (NBRW(N%) - CATAVGR(28)) ^ 2
            DEVNUMC(28) = DEVNUMC(28) + (NBCL(N%) - CATAVGC(28)) ^ 2
         CASE 29
            DEVNUMR(29) = DEVNUMR(29) + (NBRW(N%) - CATAVGR(29)) ^ 2
            DEVNUMC(29) = DEVNUMC(29) + (NBCL(N%) - CATAVGC(29)) ^ 2
         CASE 30
            DEVNUMR(30) = DEVNUMR(30) + (NBRW(N%) - CATAVGR(30)) ^ 2
            DEVNUMC(30) = DEVNUMC(30) + (NBCL(N%) - CATAVGC(30)) ^ 2
         CASE 31
            DEVNUMR(31) = DEVNUMR(31) + (NBRW(N%) - CATAVGR(31)) ^ 2
            DEVNUMC(31) = DEVNUMC(31) + (NBCL(N%) - CATAVGC(31)) ^ 2
         CASE 32
            DEVNUMR(32) = DEVNUMR(32) + (NBRW(N%) - CATAVGR(32)) ^ 2
            DEVNUMC(32) = DEVNUMC(32) + (NBCL(N%) - CATAVGC(32)) ^ 2
      END SELECT
   END SUB

   SUB DIVBY1
      NBCL(I%) = NB(I%)
      FWAVG(I%) = FW(I%)
      NBRW(I%) = 1
   END SUB

   SUB DIVBY2
      NBCL(I%) = (NB(I%) / 2)
      FWAVG(I%) = FW(I%) / 2
      NBRW(I%) = 2
   END SUB

   SUB DIVBY3
      NBCL(I%) = (NB(I%) / 3)
      FWAVG(I%) = FW(I%) / 3
      NBRW(I%) = 3
   END SUB

   SUB DIVBY4
      NBCL(I%) = (NB(I%) / 4)
      FWAVG(I%) = FW(I%) / 4
      NBRW(I%) = 4
   END SUB
```

```
SUB DIVBY5
   NBCL(I%) = (NB(I%) / 5)
   FWAVG(I%) = FW(I%) / 5
   NBRW(I%) = 5
END SUB

SUB DIVBY6
   NBCL(I%) = (NB(I%) / 6)
   FWAVG(I%) = FW(I%) / 6
   NBRW(I%) = 6
END SUB

SUB DIVBY7
   NBCL(I%) = (NB(I%) / 7)
   FWAVG(I%) = FW(I%) / 7
   NBRW(I%) = 7
END SUB

SUB LABEL
   IF M% = 1 THEN PRINT #2, " (AA)";
   IF M% = 2 THEN PRINT #2, " (AB)";
   IF M% = 3 THEN PRINT #2, " (AC)";
   IF M% = 4 THEN PRINT #2, " (AD)";
   IF M% = 5 THEN PRINT #2, " (BA)";
   IF M% = 6 THEN PRINT #2, " (BB)";
   IF M% = 7 THEN PRINT #2, " (BC)";
   IF M% = 8 THEN PRINT #2, " (BD)";
   IF M% = 9 THEN PRINT #2, " (CA)";
   IF M% = 10 THEN PRINT #2, " (CB)";
   IF M% = 11 THEN PRINT #2, " (CC)";
   IF M% = 12 THEN PRINT #2, " (CD)";
   IF M% = 13 THEN PRINT #2, " (DA)";
   IF M% = 14 THEN PRINT #2, " (DB)";
   IF M% = 15 THEN PRINT #2, " (DC)";
   IF M% = 16 THEN PRINT #2, " (DD)";
   IF M% = 17 THEN PRINT #2, " (EA)";
   IF M% = 18 THEN PRINT #2, " (EB)";
   IF M% = 19 THEN PRINT #2, " (EC)";
   IF M% = 20 THEN PRINT #2, " (ED)";
   IF M% = 21 THEN PRINT #2, " (FA)";
   IF M% = 22 THEN PRINT #2, " (FB)";
   IF M% = 23 THEN PRINT #2, " (FC)";
   IF M% = 24 THEN PRINT #2, " (FD)";
   IF M% = 25 THEN PRINT #2, " (GA)";
   IF M% = 26 THEN PRINT #2, " (GB)";
   IF M% = 27 THEN PRINT #2, " (GC)";
   IF M% = 28 THEN PRINT #2, " (GD)";
   IF M% = 29 THEN PRINT #2, " (HA)";
   IF M% = 30 THEN PRINT #2, " (HB)";
   IF M% = 31 THEN PRINT #2, " (HC)";
   IF M% = 32 THEN PRINT #2, " (HD)";
END SUB

SUB MESSAGE
   LINE (225, 270)-(410, 350), 9, B
   COLOR 11
   LOCATE 19, 33: PRINT "Input complete."
   COLOR 14
   LOCATE 20, 33: PRINT "Please wait for"
   LOCATE 21, 33: PRINT "computations..."
   COLOR 15
END SUB

SUB RESULTS
   CLS
```

```
   LINE (90, 175)-(560, 245), 9, B
   COLOR 12
   LOCATE 13, 38: PRINT "FINISHED!"
   COLOR 15
   LOCATE 14, 20: PRINT "The results have been saved in ";
   COLOR 11
   PRINT FILEO$
   COLOR 15
                            P    R    I    N    T            #    2    ,
"*************************************************************************
*******"
   PRINT #2, "*"; SPC(16); RIVER$; " RIVER, MILEMARKER DESIGNATION:";
MILE; TAB(75); "*"
   PRINT #2, "*"; SPC(33); DATE$; TAB(75); "*"
   PRINT #2, "*"; SPC(34); TIME$; TAB(75); "*"
                            P    R    I    N    T            #    2    ,
"*************************************************************************
*******"
   PRINT #2,
     PRINT #2, " CATEGORY"; TAB(16); "AVERAGE NUMBER OF"; TAB(36);
"COLUMN"; TAB(48); "AVERAGE NUMBER OF"; TAB(70); "ROW"
   PRINT #2, TAB(16); "BARGES IN COLUMN"; TAB(36); "MAXIMUM"; TAB(48);
"BARGES IN THE ROW"; TAB(68); "MAXIMUM"
   PRINT #2, TAB(16); "(AVERAGE NUMBER OF"; TAB(48); "(AVERAGE NUMBER
OF"
   PRINT #2, TAB(16); "BARGES IN COLUMN"; TAB(48); "BARGES IN THE ROW"
   PRINT #2, TAB(16); "PLUS 2 STD DEVs)"; TAB(48); "PLUS 2 STD DEVs)"
   PRINT #2,
   FOR M% = 1 TO 32
      IF M% = 5 THEN PRINT #2,
      IF M% = 9 THEN PRINT #2,
      IF M% = 13 THEN PRINT #2,
      IF M% = 17 THEN PRINT #2,
      IF M% = 21 THEN PRINT #2,
      IF M% = 25 THEN PRINT #2,
      IF M% = 29 THEN PRINT #2,
      PRINT #2, USING "##"; TAB(2); M%;
      CALL LABEL
      PRINT #2, USING "##.####"; TAB(16); CATAVGC(M%);
      PRINT #2, TAB(24); "(";
      PRINT #2, USING "##.####"; TAB(25); (CATAVGC(M%) + STDC2(M%));
      PRINT #2, TAB(32); ")";
      PRINT #2, USING "##.####"; TAB(36); MAXC(M%);
      IF MAXC(M%) < (CATAVGC(M%) + STDC2(M%)) THEN
         PRINT #2, "*";
      END IF
      PRINT #2, USING "##.####"; TAB(48); CATAVGR(M%);
      PRINT #2, TAB(56); "(";
      PRINT #2, USING "##.####"; TAB(57); (CATAVGR(M%) + STDR2(M%));
      PRINT #2, TAB(64); ")";
      PRINT #2, USING "#.####"; TAB(68); MAXR(M%);
      IF MAXR(M%) < (CATAVGR(M%) + STDR2(M%)) THEN
         PRINT #2, "*"
      ELSE
         PRINT #2, ""
      END IF
   NEXT M%
END SUB
```

Appendix IV:

Barge Distribution by Category

NOTE: **Based upon the U.S. Army Corps of Engineers barge length and width designations, there are 32 possible categories. However, it was determined by the computer program that there are actually only 24 barge categories occurring on the inland waterway system of the U.S. In addition, the program has shown there to be only 12 categories occurring on the Maysville section of the Ohio River.**

		Ohio River, Milepost 436 Flotilla Distribution					ADJUSTED
		Total Barges	Barges per Flotilla			Passages per Year	Passages per Year
			column	row	total		
1	AA	18	0.0000	0.0000	0.0000	0	**0**
2	AB	30	0.0000	0.0000	0.0000	0	**0**
3	AC	1	4.4000	1.2000	5.2800	0	**0**
4	AD	0	0.0000	0.0000	0.0000	0	**0**
5	BA	0	1.0000	1.0000	1.0000	0	**0**
6	BB	43	3.3333	1.6667	5.5556	8	**8**
7	BC	1170	3.4219	1.7188	5.8816	199	**209**
8	BD	0	3.2500	1.2500	4.0625	0	**0**
9	CA	3	0.0000	0.0000	0.0000	0	**0**
10	CB	67	0.0000	0.0000	0.0000	0	**0**
11	CC	571	3.3490	1.9688	6.5935	87	**91**
12	CD	11	0.0000	0.0000	0.0000	0	**0**
13	DA	0	0.0000	0.0000	0.0000	0	**0**
14	DB	233	5.0000	1.0000	5.0000	47	**49**
15	DC	51269	4.5837	2.8274	12.9600	3956	**4151**
16	DD	11	6.0000	2.0000	12.0000	1	**1**
17	EA	0	0.0000	0.0000	0.0000	0	**0**
18	EB	4	5.0000	1.0000	5.0000	1	**1**
19	EC	4822	4.5837	2.8274	12.9600	372	**390**
20	ED	15	0.0000	0.0000	0.0000	0	**0**
21	FA	0	0.0000	0.0000	0.0000	0	**0**
22	FB	0	1.0000	2.0000	2.0000	0	**0**
23	FC	70	3.3537	2.3159	7.7668	9	**9**
24	FD	1	0.0000	0.0000	0.0000	0	**0**
25	GA	0	0.0000	0.0000	0.0000	0	**0**
26	GB	0	0.0000	0.0000	0.0000	0	**0**
27	GC	2628	3.3884	1.9876	6.7348	390	**409**
28	GD	15	0.0000	0.0000	0.0000	0	**0**
29	HA	0	0.0000	0.0000	0.0000	0	**0**
30	HB	0	1.0000	1.0000	1.0000	0	**0**
31	HC	30	2.0000	1.7176	3.4352	9	**9**
32	HD	61	1.6667	1.0000	1.6667	37	**38**
TOTAL		61073				5114	**5367**

Appendix V:

Equivalent Static Loads Calculations

NOTE: **Calculations are performed for the 24 possible flotilla categories, but for the Maysville section of the Ohio River, only 12 categories are significant.**

THE MAYSVILLE KENTUCKY BRIDGE OVER THE OHIO RIVER
BARGE EQUIVALENT STATIC IMPACT FORCE CALCULATIONS

Barge Design Impact Velocity (fps)
West Pier: Barge V=7.0 mph (10.27 fps),
Waterway V=5.7 fps.

$V_W := (10.27 + 5.7)$

Barge Design Impact Velocity (fps)
West Pier: Barge V=7.0 mph (10.27 fps),
Waterway V=6.1 fps.

$V_E := (10.27 + 6.1)$

Hydrodynamic Coefficient

$C_H := 1.05$

Barge Row Displacement (tons): By flotilla type, i = 1,2,...,24.

T_i defines the 95.5 percentile barge tonnages.

$T_1 := 609$ $T_9 := 1868$ $T_{17} := 3046$

$T_2 := 953$ $T_{10} := 3653$ $T_{18} := 7714$

$T_3 := 4486$ $T_{11} := 421$ $T_{19} := 5315$

$T_4 := 501$ $T_{12} := 1890$ $T_{20} := 4260$

$T_5 := 1420$ $T_{13} := 2715$ $T_{21} := 6477$

$T_6 := 1226$ $T_{14} := 2643$ $T_{22} := 7497$

$T_7 := 3414$ $T_{15} := 1156$ $T_{23} := 8383$

$T_8 := 3664$ $T_{16} := 1375$ $T_{24} := 6350$

Average Flotilla Column Tonnage: Category barge tonnage times the category 95.5 percentile number of barges in a flotilla column. Catagories with zero column length do not occur on the Maysville section of the Ohio River.

$T_1 := T_1 \cdot 0.0$ $T_9 := T_9 \cdot 0.0$ $T_{17} := T_{17} \cdot 5.38$

$T_2 := T_2 \cdot 0.0$ $T_{10} := T_{10} \cdot 5.11$ $T_{18} := T_{18} \cdot 0.0$

$T_3 := T_3 \cdot 0.0$ $T_{11} := T_{11} \cdot 0.0$ $T_{19} := T_{19} \cdot 4.50$

$T_4 := T_4 \cdot 0.0$ $T_{12} := T_{12} \cdot 5.0$ $T_{20} := T_{20} \cdot 0.0$

$T_5 := T_5 \cdot 0.0$ $T_{13} := T_{13} \cdot 5.38$ $T_{21} := T_{21} \cdot 4.0$

$T_6 := T_6 \cdot 4.0$ $T_{14} := T_{14} \cdot 6.0$ $T_{22} := T_{22} \cdot 0.0$

$T_7 := T_7 \cdot 5.92$ $T_{15} := T_{15} \cdot 0.0$ $T_{23} := T_{23} \cdot 3.23$

$T_8 := T_8 \cdot 0.0$ $T_{16} := T_{16} \cdot 5.0$ $T_{24} := T_{24} \cdot 2.0$

Barge Flotilla Kinetic Energy West Tower Pier (k-ft)

$i := 1, 2 .. 24$

$$KE_{W_i} := \frac{C_H \cdot T_i \cdot 2 \cdot (V_W)^2}{2 \cdot 32.2}$$

$i := 1, 2 .. 12$

KE_{W_i}

0
0
0
0
0
$4.08 \cdot 10^4$
$1.68 \cdot 10^5$
0
0
$1.55 \cdot 10^5$
0
$7.86 \cdot 10^4$

Barge Flotilla Kinetic Energy East Tower Pier (k-ft)

$$KE_{E_i} := \frac{C_H \cdot T_i \cdot 2 \cdot (V_E)^2}{2 \cdot 32.2}$$

KE_{E_i}

0
0
0
0
0
$4.29 \cdot 10^4$
$1.77 \cdot 10^5$
0
0
$1.63 \cdot 10^5$
0
$8.26 \cdot 10^4$

$i := 13..24$

KE_{W_i}	KE_{E_i}
$1.21 \cdot 10^5$	$1.28 \cdot 10^5$
$1.32 \cdot 10^5$	$1.39 \cdot 10^5$
0	0
$5.72 \cdot 10^4$	$6.01 \cdot 10^4$
$1.36 \cdot 10^5$	$1.43 \cdot 10^5$
0	0
$1.99 \cdot 10^5$	$2.09 \cdot 10^5$
0	0
$2.15 \cdot 10^5$	$2.26 \cdot 10^5$
0	0
$2.25 \cdot 10^5$	$2.37 \cdot 10^5$
$1.06 \cdot 10^5$	$1.11 \cdot 10^5$

Barge Width Correction Factors: Using the Most Conservative Width in the Barge Catagory.

$R_{B_1} := \frac{25.70}{35}$

$R_{B_2} := \frac{33.13}{35}$

$R_{B_3} := \frac{54.00}{35}$

$R_{B_4} := \frac{55.00}{35}$

$R_{B_5} := \frac{25.00}{35}$

$R_{B_6} := \frac{33.59}{35}$

$R_{B_7} := \frac{54.00}{35}$

$R_{B_8} := \frac{59.30}{35}$

$R_{B_9} := \frac{26.98}{35}$

$R_{B_{10}} := \frac{54.00}{35}$

$R_{B_{11}} := \frac{60.00}{35}$

$R_{B_{12}} := \frac{26.74}{35}$

$R_{B_{13}} := \frac{37.49}{35}$

$R_{B_{14}} := \frac{54.00}{35}$

$R_{B_{15}} := \frac{25.00}{35}$

$R_{B_{16}} := \frac{26.00}{35}$

$R_{B_{17}} := \frac{43.23}{35}$

$R_{B_{18}} := \frac{72.00}{35}$

$R_{B_{19}} := \frac{54.00}{35}$

$R_{B_{20}} := \frac{62.69}{35}$

$R_{B_{21}} := \frac{54.00}{35}$

$R_{B_{22}} := \frac{56.64}{35}$

$R_{B_{23}} := \frac{54.00}{35}$

$R_{B_{24}} := \frac{55.82}{35}$

Barge Damage Depth a_{BW} West Tower Pier (ft.)

Barge Damage Depth a_{BE} East Tower Pier (ft.)

$i := 1..24$

$$a_{BW_i} := \frac{\left[\left(1 + \frac{KE_{W_i}}{5672}\right)^{\frac{1}{2}} - 1\right] \cdot 10.2}{R_{B_i}}$$

$$a_{BE_i} := \frac{\left[\left(1 + \frac{KE_{E_i}}{5672}\right)^{\frac{1}{2}} - 1\right] \cdot 10.2}{R_{B_i}}$$

a_{BW_i}	a_{BE_i}
0	0
0	0
0	0
0	0
0	0
19.79	20.46
29.98	30.87
0	0
0	0
28.6	29.45
0	0
38.11	39.31
35.56	36.64
25.95	26.73
0	0
31.98	33.02
33.06	34.05
0	0
33.09	34.06
0	0
34.67	35.68
0	0
35.57	36.6
21.93	22.61

Barge Equivalent Static Impact Force P_{BW} for the West Tower Pier (kips)

Barge Equivalent Static Impact Force P_{BE} for the East Tower Pier (kips)

$$P_{BW_i} := \left(1349 + 110 \cdot a_{BW_i}\right) \cdot R_{B_i}$$

$$P_{BE_i} := \left(1349 + 110 \cdot a_{BE_i}\right) \cdot R_{B_i}$$

$P_{BW_1} := 0.0$

$P_{BW_2} := 0.0$

$P_{BW_3} := 0.0$

$P_{BW_4} := 0.0$

$P_{BW_5} := 0.0$

$P_{BW_8} := 0.0$

$P_{BW_9} := 0.0$

$P_{BW_{11}} := 0.0$

$P_{BW_{15}} := 0.0$

$P_{BW_{18}} := 0.0$

$P_{BW_{20}} := 0.0$

$P_{BW_{22}} := 0.0$

$P_{BE_1} := 0.0$

$P_{BE_2} := 0.0$

$P_{BE_3} := 0.0$

$P_{BE_4} := 0.0$

$P_{BE_5} := 0.0$

$P_{BE_8} := 0.0$

$P_{BE_9} := 0.0$

$P_{BE_{11}} := 0.0$

$P_{BE_{15}} := 0.0$

$P_{BE_{18}} := 0.0$

$P_{BE_{20}} := 0.0$

$P_{BE_{22}} := 0.0$

P_{BW_i}

0
0
0
0
0
$3.38 \cdot 10^3$
$7.17 \cdot 10^3$
0
0
$6.94 \cdot 10^3$
0
$4.23 \cdot 10^3$
$5.64 \cdot 10^3$
$6.48 \cdot 10^3$
0
$3.61 \cdot 10^3$
$6.16 \cdot 10^3$
0
$7.7 \cdot 10^3$
0
$7.97 \cdot 10^3$
0
$8.12 \cdot 10^3$
$6 \cdot 10^3$

P_{BE_i}

0
0
0
0
0
$3.45 \cdot 10^3$
$7.32 \cdot 10^3$
0
0
$7.08 \cdot 10^3$
0
$4.33 \cdot 10^3$
$5.76 \cdot 10^3$
$6.62 \cdot 10^3$
0
$3.7 \cdot 10^3$
$6.29 \cdot 10^3$
0
$7.86 \cdot 10^3$
0
$8.14 \cdot 10^3$
0
$8.29 \cdot 10^3$
$6.12 \cdot 10^3$

BARGE EQUIVALENT STATIC IMPACT FORCE CALCULATIONS FOR A SINGLE FREE DRIFTING BARGE

Barge Design Impact Velocity (fps)

West Pier: Barge V=0.0 mph
& Waterway V=6.8 fps

East Pier: Barge V=0.0 mph
& Waterway V=7.1 fps

$V_W := 6.8$

$V_E := 7.1$

Single Free-Drifting Barge Loaded Tonnage (tons):

$T_i := 6477$

Barge Kinetic Energy West Tower Pier (k-ft)

$i := 1 \qquad R_{B_i} := 1.53$

$$KE_{W_i} := \frac{C_H \cdot T_i \cdot 2 \cdot V_W^2}{2 \cdot 32.2}$$

$KE_{W_i} = 9.77 \cdot 10^3$

Barge Kinetic Energy East Tower Pier (k-ft)

$i := 1 \qquad R_{B_i} := 1.53$

$$KE_{E_i} := \frac{C_H \cdot T_i \cdot 2 \cdot V_E^2}{2 \cdot 32.2}$$

$KE_{E_i} = 1.06 \cdot 10^4$

Barge Damage Depth a_{BW} West Tower Pier (ft.)

$$a_{BW_i} := \frac{\left[\left(1 + \frac{KE_{W_i}}{5672}\right)^{\frac{1}{2}} - 1\right] \cdot 10.2}{R_{B_i}}$$

$a_{BW_i} = 4.33$

Barge Damage Depth a_{BE} East Tower Pier (ft.)

$$a_{BE_i} := \frac{\left[\left(1 + \frac{KE_{E_i}}{5672}\right)^{\frac{1}{2}} - 1\right] \cdot 10.2}{R_{B_i}}$$

$a_{BE_i} = 4.64$

Barge Equivalent Static Impact Force P_{BW} for the West Tower Pier (kips)

$$P_{BW_i} := \left(1349 + 110 \cdot a_{BW_i}\right) \cdot R_{B_i}$$

$P_{BW_i} = 2.79 \cdot 10^3$

Barge Equivalent Static Impact Force P_{BE} for the East Tower Pier (kips)

$$P_{BE_i} := \left(1349 + 110 \cdot a_{BE_i}\right) \cdot R_{B_i}$$

$P_{BE_i} = 2.85 \cdot 10^3$