Lesson 6 Homework \_ISM6930

1. Rewrite this program using arrays:

**data** prob1;

input (HT1-HT5)(**2.**) (WT1-WT5)(**3.**);

dens1=WT1/HT1\*\***2**;

dens2=WT2/HT2\*\***2**;

dens3=WT3/HT3\*\***2**;

dens4=WT4/HT4\*\***2**;

dens5=WT5/HT5\*\***2**;

datalines;

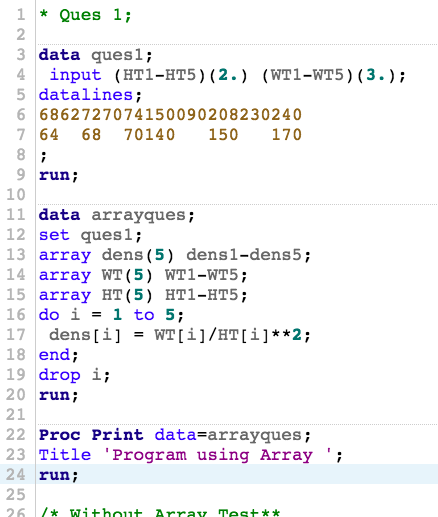
DATALINES;

6862727074150090208230240

64 68 70140 150 170

;

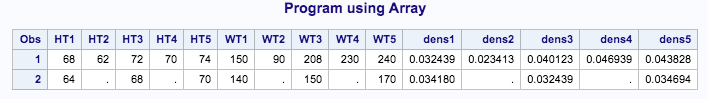
**Code:**



**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \* Ques 1;  74  75 data ques1;  76 input (HT1-HT5)(2.) (WT1-WT5)(3.);  77 datalines;    NOTE: The data set WORK.QUES1 has 2 observations and 10 variables.  NOTE: DATA statement used (Total process time):  real time 0.01 seconds  cpu time 0.01 seconds    80 ;    81 run;  82  83 data arrayques;  84 set ques1;  85 array dens(5) dens1-dens5;  86 array WT(5) WT1-WT5;  87 array HT(5) HT1-HT5;  88 do i = 1 to 5;  89 dens[i] = WT[i]/HT[i]\*\*2;  90 end;  91 drop i;  92 run;    NOTE: Missing values were generated as a result of performing an operation on missing values.  Each place is given by: (Number of times) at (Line):(Column).  2 at 89:17 2 at 89:23  NOTE: There were 2 observations read from the data set WORK.QUES1.  NOTE: The data set WORK.ARRAYQUES has 2 observations and 15 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds      93  94 Proc Print data=arrayques;  95 Title 'Program using Array ';  96 run;    NOTE: There were 2 observations read from the data set WORK.ARRAYQUES.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.06 seconds  cpu time 0.06 seconds      97  98 /\* Without Array Test\*\*  99  100 data prob1;  101 input (HT1-HT5)(2.) (WT1-WT5)(3.);  102 dens1=WT1/HT1\*\*2;  103 dens2=WT2/HT2\*\*2;  104 dens3=WT3/HT3\*\*2;  105 dens4=WT4/HT4\*\*2;  106 dens5=WT5/HT5\*\*2;  107 datalines;  108 6862727074150090208230240  109 64 68 70140 150 170  110 ;  111 run;  112  113 Proc Print data=prob1;  114 run;\*/  115  116 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  129 |

**Results:**



1. Rewrite this program using arrays:

**DATA** OLDMISS;

INPUT A B C X1-X3 Y1-Y3;

IF A=**999** then A=**.**;

IF B=**999** then B=**.**;

IF C=**999** then C=**.**;

IF X1=**999** then X1=**.**;

IF X2=**999** then X2=**.**;

IF X3=**999** then X3=**.**;

IF Y1=**999** then Y1=**.**;

IF Y2=**999** then Y2=**.**;

IF Y3=**999** then Y3=**.**;

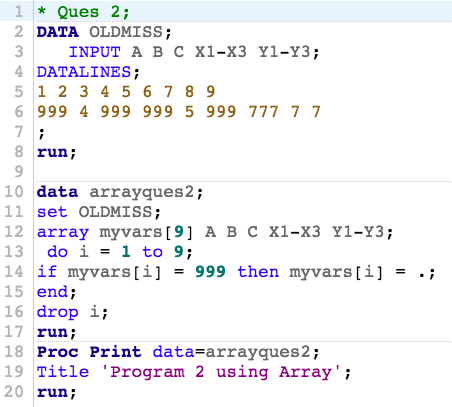
DATALINES;

1 2 3 4 5 6 7 8 9

999 4 999 999 5 999 777 7 7

;

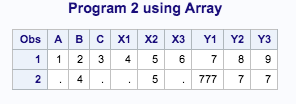
**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \* Ques 2;  74 DATA OLDMISS;  75 INPUT A B C X1-X3 Y1-Y3;  76 DATALINES;    NOTE: The data set WORK.OLDMISS has 2 observations and 9 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds    79 ;    80 run;  81  82 data arrayques2;  83 set OLDMISS;  84 array myvars[9] A B C X1-X3 Y1-Y3;  85 do i = 1 to 9;  86 if myvars[i] = 999 then myvars[i] = .;  87 end;  88 drop i;  89 run;    NOTE: There were 2 observations read from the data set WORK.OLDMISS.  NOTE: The data set WORK.ARRAYQUES2 has 2 observations and 9 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.02 seconds      90 Proc Print data=arrayques2;  91 Title 'Program 2 using Array';  92 run;    NOTE: There were 2 observations read from the data set WORK.ARRAYQUES2.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.05 seconds  cpu time 0.05 seconds      93  94 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  107 |

**Results:**

****

1. You are given the SAS dataset SPEED, created by running the following program. Create a new dataset SPEED2 from SPEED, with some new variables. The new variables LX1-LX5 are the natural (base e) logs of the variables X1-X5, and the variables SY1-SY3 are the square roots of the variables Y1-Y3. Use arrays to create the new variables. NOTE: LOG\_X=LOG(X)

**DATA** SPEED;

INPUT X1-X5 Y1-Y3;

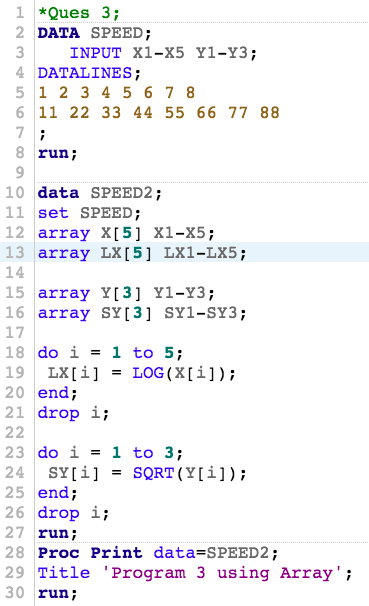
DATALINES;

1 2 3 4 5 6 7 8

11 22 33 44 55 66 77 88

;

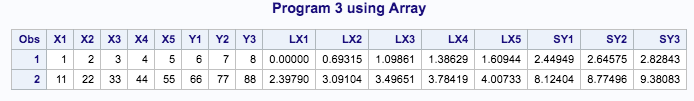
**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \*Ques 3;  74 DATA SPEED;  75 INPUT X1-X5 Y1-Y3;  76 DATALINES;    NOTE: The data set WORK.SPEED has 2 observations and 8 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.01 seconds    79 ;    80 run;  81  82 data SPEED2;  83 set SPEED;  84 array X[5] X1-X5;  85 array LX[5] LX1-LX5;  86  87 array Y[3] Y1-Y3;  88 array SY[3] SY1-SY3;  89  90 do i = 1 to 5;  91 LX[i] = LOG(X[i]);  92 end;  93 drop i;  94  95 do i = 1 to 3;  96 SY[i] = SQRT(Y[i]);  97 end;  98 drop i;  99 run;    NOTE: There were 2 observations read from the data set WORK.SPEED.  NOTE: The data set WORK.SPEED2 has 2 observations and 16 variables.  NOTE: DATA statement used (Total process time):  real time 0.01 seconds  cpu time 0.01 seconds      100 Proc Print data=SPEED2;  101 Title 'Program 3 using Array';  102 run;    NOTE: There were 2 observations read from the data set WORK.SPEED2.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.06 seconds  cpu time 0.07 seconds      103  104 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  117 |

**Results:**

****

1. We have the dataset FROG that looks like this:

ID X1 X2 X3 X4 X5 Y1 Y2 Y3 Y4 Y5

01 4 5 5 7 3 1 7 3 6 8

02 8 7 8 6 7 5 4 3 5 6

We want a data set that has an observation for each subject (ID) at each time interval (X1 represents X at time 1, etc.). Write a program, using arrays, to accomplish this objective. The new data set (TOAD) should look like:

ID Time X Y

01 1 4 1

01 2 5 7

01 3 5 3

01 4 7 6 ETC.

Run the program to create the data set FROG:

**DATA** FROG;

INPUT ID X1-X5 Y1-Y5;

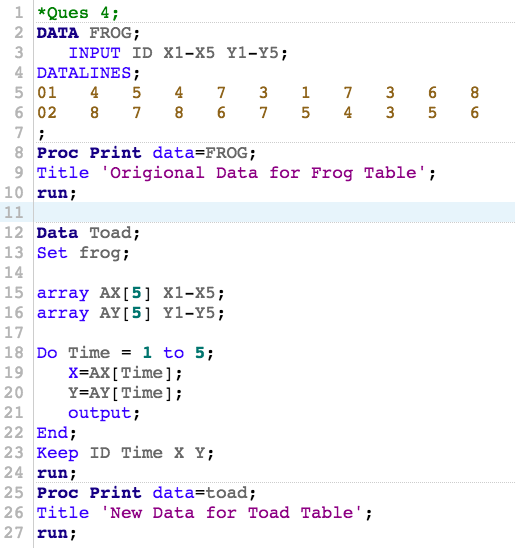
DATALINES;

01 4 5 4 7 3 1 7 3 6 8

02 8 7 8 6 7 5 4 3 5 6

;

**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \*Ques 4;  74 DATA FROG;  75 INPUT ID X1-X5 Y1-Y5;  76 DATALINES;    NOTE: The data set WORK.FROG has 2 observations and 11 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds    79 ;    80 Proc Print data=FROG;  81 Title 'Origional Data for Frog Table';  82 run;    NOTE: There were 2 observations read from the data set WORK.FROG.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.06 seconds  cpu time 0.07 seconds      83 Data Toad;  84 Set frog;  85  86 array AX[5] X1-X5;  87 array AY[5] Y1-Y5;  88  89 Do Time = 1 to 5;  90 X=AX[Time];  91 Y=AY[Time];  92 output;  93 End;  94 Keep ID Time X Y;  95 run;    NOTE: There were 2 observations read from the data set WORK.FROG.  NOTE: The data set WORK.TOAD has 10 observations and 4 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds      96 Proc Print data=toad;  97 Title 'New Data for Toad Table';  98 run;    NOTE: There were 10 observations read from the data set WORK.TOAD.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.04 seconds  cpu time 0.05 seconds      99  100 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  113 |

**Results:**



1. We have a dataset (called STATE) that contains ID variable and up to five states (two-letter codes) where an individual may have visited last year. Three observations from this data set are shown:

ID STATE1 STATE2 STATE3 STATE4 STATE5

1 NY NJ PA TX GA

2 NJ NY CA XX XX

3 PA XX XX XX XX

As you see “XX” was used as a missing value. Write a program to (a) read these records and replace the “XX” with blanks, and (b) compute the frequency counts showing how many people visited each state. Present the frequency list in decreasing order of frequency (use the ORDER-FREQ option in PROC FREQ). Run the following program to create the data set STATE:

**DATA** STATE;

INFORMAT STATE1-STATE5 $2.;

INPUT ID STATE1-STATE5;

DATALINES;

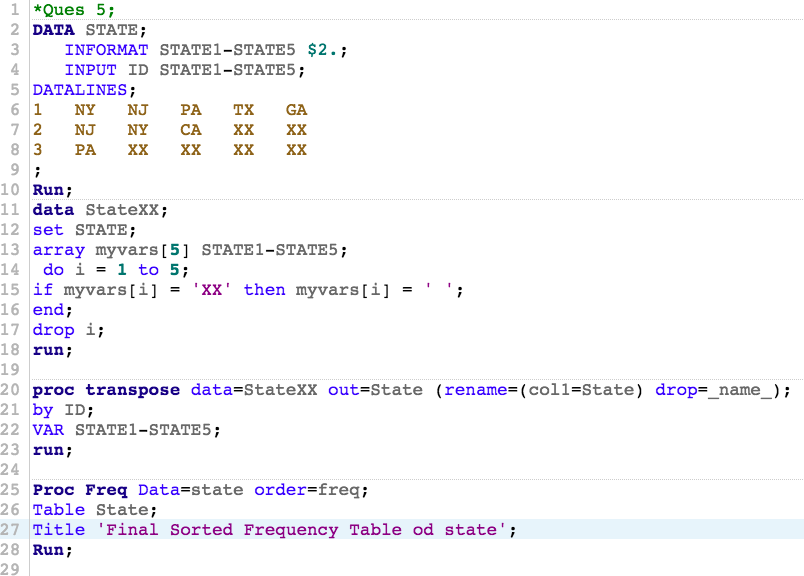
1 NY NJ PA TX GA

2 NJ NY CA XX XX

3 PA XX XX XX XX

;

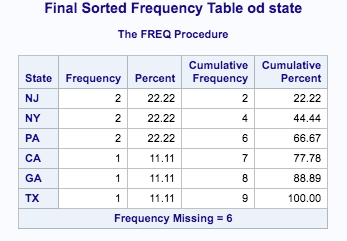
**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \*Ques 5;  74 DATA STATE;  75 INFORMAT STATE1-STATE5 $2.;  76 INPUT ID STATE1-STATE5;  77 DATALINES;    NOTE: The data set WORK.STATE has 3 observations and 6 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds    81 ;    82 Run;  83 data StateXX;  84 set STATE;  85 array myvars[5] STATE1-STATE5;  86 do i = 1 to 5;  87 if myvars[i] = 'XX' then myvars[i] = ' ';  88 end;  89 drop i;  90 run;    NOTE: There were 3 observations read from the data set WORK.STATE.  NOTE: The data set WORK.STATEXX has 3 observations and 6 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds      91  92 proc transpose data=StateXX out=State (rename=(col1=State) drop=\_name\_);  93 by ID;  94 VAR STATE1-STATE5;  95 run;    NOTE: There were 3 observations read from the data set WORK.STATEXX.  NOTE: The data set WORK.STATE has 15 observations and 2 variables.  NOTE: PROCEDURE TRANSPOSE used (Total process time):  real time 0.00 seconds  cpu time 0.01 seconds      96  97 Proc Freq Data=state order=freq;  98 Table State;  99 Title 'Final Sorted Frequency Table od state';  100 Run;    NOTE: There were 15 observations read from the data set WORK.STATE.  NOTE: PROCEDURE FREQ used (Total process time):  real time 0.07 seconds  cpu time 0.07 seconds      101  102  103  104  105  106  107 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  120 |

**Results:**



1. You have inherited an old SAS program (shown here) and want to convert it to one using explicit array subscripts. Rewrite the program to do this.

**DATA** NEW;

SET BLAH;

ARRAY JUNK(J) X1-X5 Y1-Y5 Z1-Z5;

DO OVER JUNK;

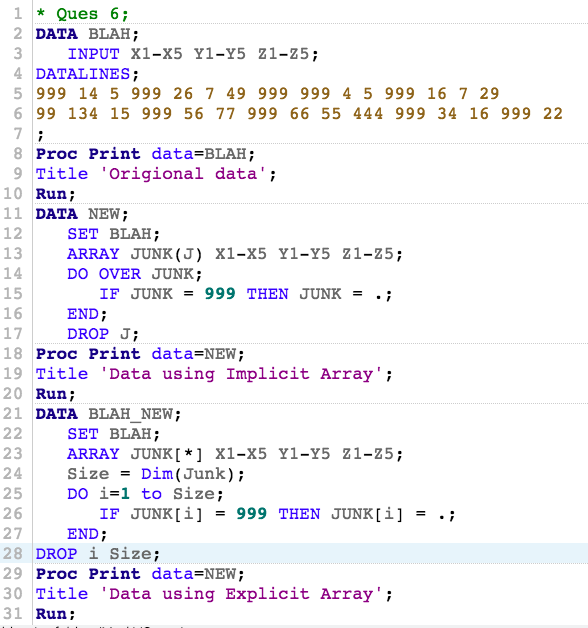
IF JUNK = **999** THEN JUNK = **.**;

END;

DROP J;

**RUN**;

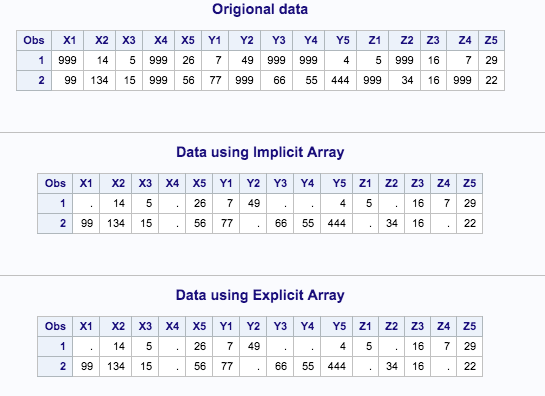
**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \* Ques 6;  74 DATA BLAH;  75 INPUT X1-X5 Y1-Y5 Z1-Z5;  76 DATALINES;    NOTE: The data set WORK.BLAH has 2 observations and 15 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds    79 ;    80 Proc Print data=BLAH;  81 Title 'Origional data';  82 Run;    NOTE: There were 2 observations read from the data set WORK.BLAH.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.05 seconds  cpu time 0.05 seconds      83 DATA NEW;  84 SET BLAH;  85 ARRAY JUNK(J) X1-X5 Y1-Y5 Z1-Z5;  86 DO OVER JUNK;  87 IF JUNK = 999 THEN JUNK = .;  88 END;  89 DROP J;    NOTE: There were 2 observations read from the data set WORK.BLAH.  NOTE: The data set WORK.NEW has 2 observations and 15 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.01 seconds    90 Proc Print data=NEW;    91 Title 'Data using Implicit Array';  92 Run;    NOTE: There were 2 observations read from the data set WORK.NEW.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.03 seconds  cpu time 0.03 seconds      93 DATA BLAH\_NEW;  94 SET BLAH;  95 ARRAY JUNK[\*] X1-X5 Y1-Y5 Z1-Z5;  96 Size = Dim(Junk);  97 DO i=1 to Size;  98 IF JUNK[i] = 999 THEN JUNK[i] = .;  99 END;  100 DROP i Size;    NOTE: There were 2 observations read from the data set WORK.BLAH.  NOTE: The data set WORK.BLAH\_NEW has 2 observations and 15 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.01 seconds    101 Proc Print data=NEW;    102 Title 'Data using Explicit Array';  103 Run;    NOTE: There were 2 observations read from the data set WORK.NEW.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.03 seconds  cpu time 0.04 seconds      104  105  106  107 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  120 |

**Results:**



1. Refer to HOCKEY data. Write a SAS program which calculates the number of games won, lost, and tied up to and including the current observation. Print the dataset with an appropriate format for the date. Don’t forget to change the score of the final game to Boston College 5, Ohio State 2 (do this in your code, don’t change the original file). The first few lines of output should be similar to this:

DATE TEAM CITY STATE OSU OPP W L T

10/10/97 Toronto Columbus Ohio 5 0 1 0 0

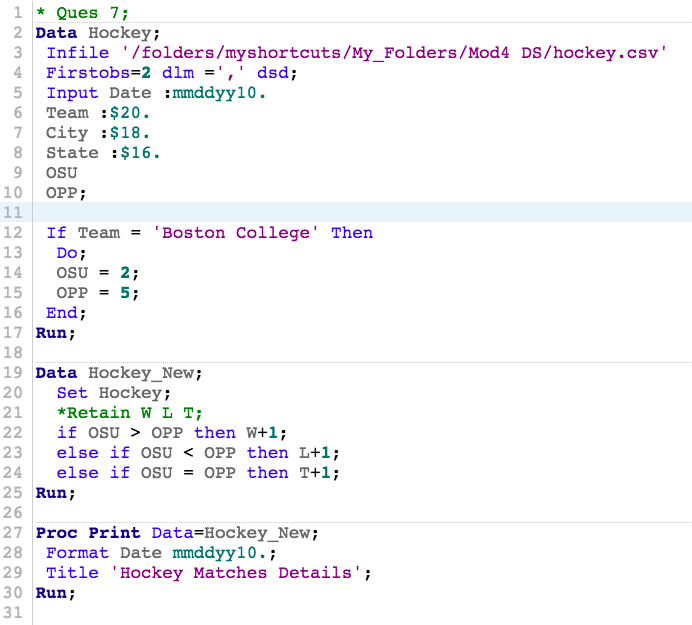
10/18/97 Miami Oxford Ohio 0 3 1 1 0

10/24/97 Merrimack Columbus Ohio 2 7 1 2 0

10/26/97 Merrimack Columbus Ohio 5 3 2 2 0

10/31/97 Clarkson Potsdam New York 1 1 2 2 1

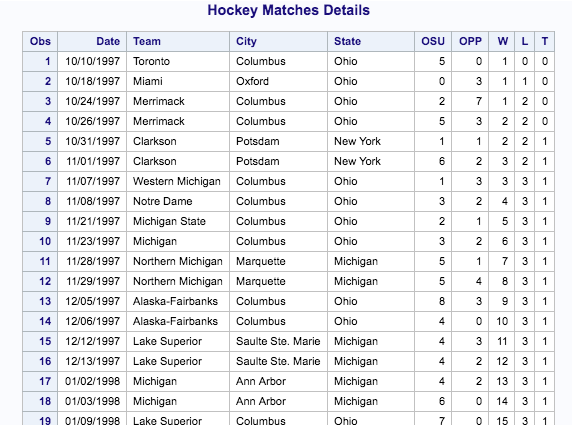
**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \* Ques 7;  74 Data Hockey;  75 Infile '/folders/myshortcuts/My\_Folders/Mod4 DS/hockey.csv'  76 Firstobs=2 dlm =',' dsd;  77 Input Date :mmddyy10.  78 Team :$20.  79 City :$18.  80 State :$16.  81 OSU  82 OPP;  83  84 If Team = 'Boston College' Then  85 Do;  86 OSU = 2;  87 OPP = 5;  88 End;  89  90 Run;    NOTE: The infile '/folders/myshortcuts/My\_Folders/Mod4 DS/hockey.csv' is:  Filename=/folders/myshortcuts/My\_Folders/Mod4 DS/hockey.csv,  Owner Name=root,Group Name=vboxsf,  Access Permission=-rwxrwx---,  Last Modified=01Mar2018:22:26:10,  File Size (bytes)=1701    NOTE: 36 records were read from the infile '/folders/myshortcuts/My\_Folders/Mod4 DS/hockey.csv'.  The minimum record length was 35.  The maximum record length was 56.  NOTE: The data set WORK.HOCKEY has 36 observations and 6 variables.  NOTE: DATA statement used (Total process time):  real time 0.02 seconds  cpu time 0.01 seconds      91  92 Data Hockey\_New;  93 Set Hockey;  94 \*Retain W L T;  95 if OSU > OPP then W+1;  96 else if OSU < OPP then L+1;  97 else if OSU = OPP then T+1;  98 Run;    NOTE: There were 36 observations read from the data set WORK.HOCKEY.  NOTE: The data set WORK.HOCKEY\_NEW has 36 observations and 9 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.01 seconds      99  100 Proc Print Data=Hockey\_New;  101 Format Date mmddyy10.;  102 Title 'Hockey Matches Details';  103 Run;    NOTE: There were 36 observations read from the data set WORK.HOCKEY\_NEW.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.15 seconds  cpu time 0.13 seconds      104  105  106 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  119 |

**Results:**



.

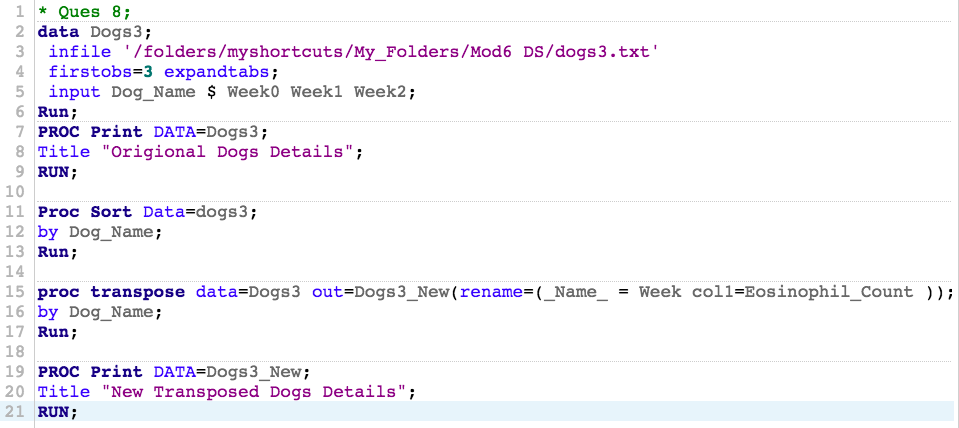
.

.



1. Refer to the DOGS3 data. Write a SAS program which creates a dataset using the INFILE statement. Then, create a new dataset which contains three variables: the name of the dog, the week of the measurement, and the eosinophil count in that week. There should be 75 observations in the new dataset. Print both datasets.

**Code:**

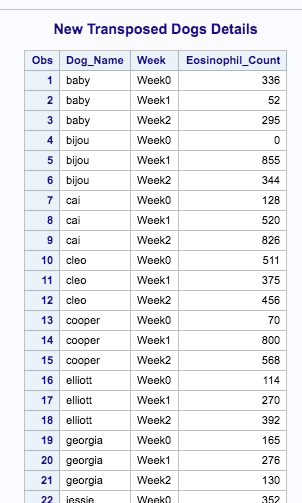
****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \* Ques 8;  74 data Dogs3;  75 infile '/folders/myshortcuts/My\_Folders/Mod6 DS/dogs3.txt'  76 firstobs=3 expandtabs;  77 input Dog\_Name $ Week0 Week1 Week2;  78 Run;    NOTE: The infile '/folders/myshortcuts/My\_Folders/Mod6 DS/dogs3.txt' is:  Filename=/folders/myshortcuts/My\_Folders/Mod6 DS/dogs3.txt,  Owner Name=root,Group Name=vboxsf,  Access Permission=-rwxrwx---,  Last Modified=27Mar2018:01:46:49,  File Size (bytes)=918    NOTE: 25 records were read from the infile '/folders/myshortcuts/My\_Folders/Mod6 DS/dogs3.txt'.  The minimum record length was 32.  The maximum record length was 32.  NOTE: The data set WORK.DOGS3 has 25 observations and 4 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds      79 PROC Print DATA=Dogs3;  80 Title "Origional Dogs Details";  81 RUN;    NOTE: There were 25 observations read from the data set WORK.DOGS3.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.07 seconds  cpu time 0.08 seconds      82  83 Proc Sort Data=dogs3;  84 by Dog\_Name;  85 Run;    NOTE: There were 25 observations read from the data set WORK.DOGS3.  NOTE: The data set WORK.DOGS3 has 25 observations and 4 variables.  NOTE: PROCEDURE SORT used (Total process time):  real time 0.01 seconds  cpu time 0.00 seconds      86  87 proc transpose data=Dogs3 out=Dogs3\_New(rename=(\_Name\_ = Week col1=Eosinophil\_Count ));  88 by Dog\_Name;  89 Run;    NOTE: There were 25 observations read from the data set WORK.DOGS3.  NOTE: The data set WORK.DOGS3\_NEW has 75 observations and 3 variables.  NOTE: PROCEDURE TRANSPOSE used (Total process time):  real time 0.01 seconds  cpu time 0.00 seconds      90  91 PROC Print DATA=Dogs3\_New;  92 Title "New Transposed Dogs Details";  93 RUN;    NOTE: There were 75 observations read from the data set WORK.DOGS3\_NEW.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.13 seconds  cpu time 0.14 seconds      94  95 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  108 |

**Results:**

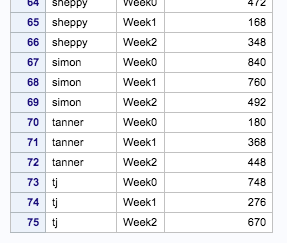
****

****

**.**

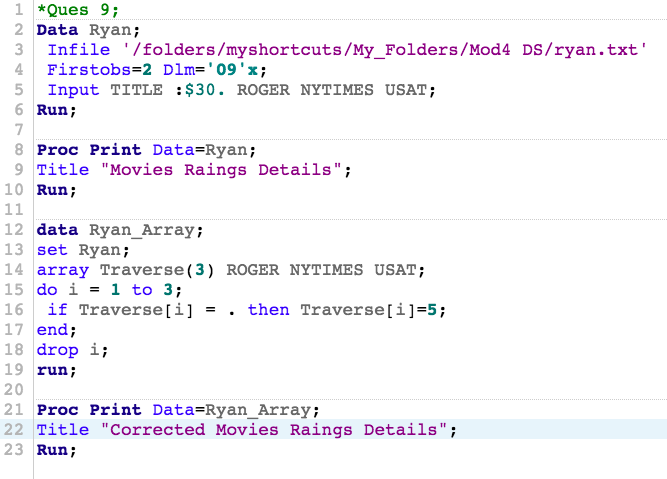
**.**

**.**

****

1. Refer to the RYAN data. Many of the ratings are missing. Suppose that a statistician wants to perform a procedure which does not allow missing values. He decides that one acceptable way to do the analysis is to replace each missing rating with the neutral rating (5). Write a SAS program which uses an array to replace all of the missing values for movie ratings with scores of five. Print the corrected dataset.

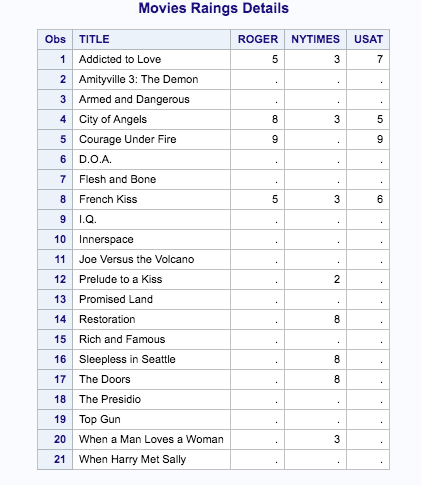
**Code:**

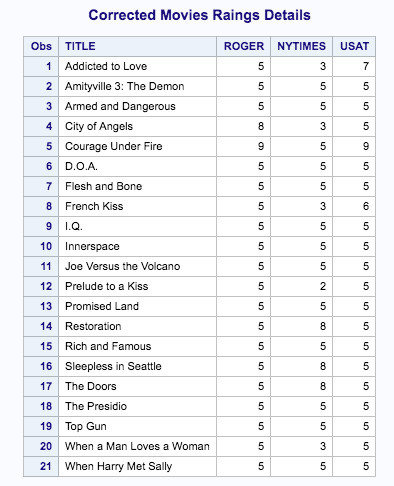
****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \*Ques 9;  74 Data Ryan;  75 Infile '/folders/myshortcuts/My\_Folders/Mod4 DS/ryan.txt'  76 Firstobs=2 Dlm='09'x;  77 Input TITLE :$30. ROGER NYTIMES USAT;  78 Run;    NOTE: The infile '/folders/myshortcuts/My\_Folders/Mod4 DS/ryan.txt' is:  Filename=/folders/myshortcuts/My\_Folders/Mod4 DS/ryan.txt,  Owner Name=root,Group Name=vboxsf,  Access Permission=-rwxrwx---,  Last Modified=02Mar2018:04:17:15,  File Size (bytes)=541    NOTE: 21 records were read from the infile '/folders/myshortcuts/My\_Folders/Mod4 DS/ryan.txt'.  The minimum record length was 13.  The maximum record length was 32.  NOTE: The data set WORK.RYAN has 21 observations and 4 variables.  NOTE: DATA statement used (Total process time):  real time 0.01 seconds  cpu time 0.01 seconds      79  80 Proc Print Data=Ryan;  81 Title "Movies Raings Details";  82 Run;    NOTE: There were 21 observations read from the data set WORK.RYAN.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.08 seconds  cpu time 0.08 seconds      83  84 data Ryan\_Array;  85 set Ryan;  86 array Traverse(3) ROGER NYTIMES USAT;  87 do i = 1 to 3;  88 if Traverse[i] = . then Traverse[i]=5;  89 end;  90 drop i;  91 run;    NOTE: There were 21 observations read from the data set WORK.RYAN.  NOTE: The data set WORK.RYAN\_ARRAY has 21 observations and 4 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.01 seconds      92  93 Proc Print Data=Ryan\_Array;  94 Title "Corrected Movies Raings Details";  95 Run;    NOTE: There were 21 observations read from the data set WORK.RYAN\_ARRAY.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.06 seconds  cpu time 0.06 seconds      96  97 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  110 |

**Results:**

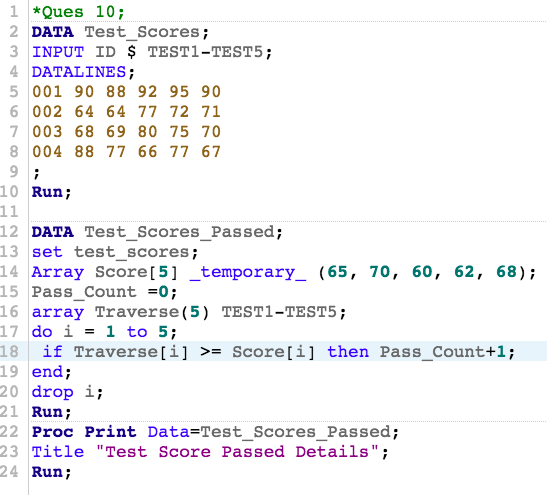




1. The passing score on each of five tests is 65, 70, 60, 62, and 68. Using the data here, use a temporary array to count the number of tests passed by each student.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test 1** | **Test 2** | **Test 3** | **Test 4** | **Test 5** |
| 001 | 90 | 88 | 92 | 95 | 90 |
| 002 | 64 | 64 | 77 | 72 | 71 |
| 003 | 68 | 69 | 80 | 75 | 70 |
| 004 | 88 | 77 | 66 | 77 | 67 |

**Code:**

****

**Log:**

|  |
| --- |
| 1 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  72  73 \*Ques 10;  74 DATA Test\_Scores;  75 INPUT ID $ TEST1-TEST5;  76 DATALINES;    NOTE: The data set WORK.TEST\_SCORES has 4 observations and 6 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds    81 ;    82 Run;  83  84 DATA Test\_Scores\_Passed;  85 set test\_scores;  86 Array Score[5] \_temporary\_ (65, 70, 60, 62, 68);  87 Pass\_Count =0;  88 array Traverse(5) TEST1-TEST5;  89 do i = 1 to 5;  90 if Traverse[i] >= Score[i] then Pass\_Count+1;  91 end;  92 drop i;  93 Run;    NOTE: There were 4 observations read from the data set WORK.TEST\_SCORES.  NOTE: The data set WORK.TEST\_SCORES\_PASSED has 4 observations and 7 variables.  NOTE: DATA statement used (Total process time):  real time 0.00 seconds  cpu time 0.00 seconds      94 Proc Print Data=Test\_Scores\_Passed;  95 Title "Test Score Passed Details";  96 Run;    NOTE: There were 4 observations read from the data set WORK.TEST\_SCORES\_PASSED.  NOTE: PROCEDURE PRINT used (Total process time):  real time 0.07 seconds  cpu time 0.08 seconds      97  98 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;  111 |

**Results:**

