```
/*****************************************************************************
********/
/* Program Name:     pulkit.jain_HW13.sas                                   */
/* Program Location: C:\Users\Pulkit
Jain\Documents\sasuniversityedition\myfolders\assign13 */
/* Date Created:     11/12/2017                                             */
/* Author:              Pulkit Jain                                     */
/* Purpose:             Assignment 13, using arrays & variable lists
*/
/*****************************************************************************
********/

/* Create two libname statements;              */
/* Assign library to locaion of hw data with access only; */
/* Assign another library with read and write access;     */

libname hw_data '/folders/myfolders/hw_data' access=readonly;
libname pulkit13 '/folders/myfolders/assign13';

/* Specify a fileref to designate output of pdf */

filename HW13 '/folders/myfolders/assign13/pulkit.jain_HW13_output.pdf';

/* 2 Create narrow dataset for scholarship funds */

data work.student_funds (keep= student_id i fund_code);
* Read only required variables;
   set hw_data.scholarships (drop = name amount: major);
   * create array reference for fund variable;
   array fund{*} $4 fund:;
   length i 4;
   * soft code the loop for automation;
   do i = 1 to dim(fund);
        if fund{i} ne . then do;
            i = i;
            fund_code = fund{i};
            output;
        end;
   end;
run;

/* 3 Sort so as to merge with fund_data data set */

PROC SORT data = work.student_funds;
     by fund_code;
run;


/* 4 Sort fund_data and save in the temporary library*/

PROC SORT data = hw_data.fund_data out = work.fund_data_sorted;
     by fund_code;
run;

/*  5 Merge the two datasets by fund code*/
data work.fund_types;
    merge work.student_funds(in = a) work.fund_data_sorted (in = b);
    by fund_code;
    drop fund_name;
    * only keep observations which are present in student_funds data set;
    if a =1;
run;

/* 6 Transform this data set back into a wide data set    */

* sort before the transpose;
PROC SORT data = fund_types out = fund_types_sorted;
      by student_id;
run;

proc transpose
       data = work.fund_types_sorted
       out=rotate2 (drop = _name_ _label_)
       prefix = Fund_Type;
      * declare variable to categorise data in row, columns;
      by student_id;
      id i;
      * declare variable to fill in the cells;
      var category;
run;

* arrange the variables in the data alphabetically;
data rotate2_arranged;
      retain student_id Fund_Type1-Fund_Type10;
      set rotate2;
run;

/* 7 Merge the Rotate2 data set with scholarships data by student id*/


data work.fund_types_extended (drop = i);
    merge hw_data.scholarships work.rotate2_arranged;
    by student_id;
    * create two array references and two variables for aid received;
      array aid_amount{*} amount:;
      array aid_name{*} Fund_Type:;
      int_aid = 0;
      ath_aid = 0;
      * loop to add Internal & Athletic aid received;
      do i = 1 to dim(aid_name);
            if aid_name{i} = 'Internal' then int_aid = sum(int_aid, aid_amount{i});
            if aid_name{i} = 'Athletic' then ath_aid = sum(ath_aid,aid_amount{i});
      end;
    tot_aid = sum(of amount:);
    * create labels for variables;
      label tot_aid = "Total Aid"
             int_aid = "Internal Scholarships"
             ath_aid = "Athletic Scholarships"
             major = "Maj_Code";
run;

/* 8 Print the descriptor and data portion of final data set*/

ods pdf file = HW13;

PROC CONTENTS data = work.fund_types_extended order=varnum;
run;

PROC PRINT data = work.fund_types_extended label noobs;
      var student_id   name major int_aid ath_aid tot_aid;
```

```
run;

ods pdf close;
ods listing;
```