Total Points: 30/30

```
/*
Name: Orysya Stus
Due Date: July 25 2016
*/

/*
Exercise 1.1. Create a SAS data set that is based on HEARING.SAS7BDAT.
The information about downloading the
testing data sets can be found in the Preface.
1. In this data set, create a variable M_INCOME (monthly income) based on
the variable INCOME (yearly income).
2. Create another variable, HEARING_INFO, which is based on the variable
HEARING. If the HEARING variable contains
missing values, HEARING_INFO will be assigned with value 1; otherwise,
HEARING_INFO will be assigned with value 0.
3. Label the variable M_INCOME with “monthly income”.
4. Format the M_INCOME variable with the FRACT9. format. Note: You might
need to check the SAS documentation for
this format. If you have difficulty finding this document, you can search
“Fractw. Format SAS” using your preferred
online search engine.
5. In the resulting data set, you need to keep the ID, HEARING,
HEARING_INFO, INCOME, and M_INCOME variables.
Once this data set is created, use PROC FREQ to create a two-way
contingency table for the variables HEARING and
HEARING_INFO to confirm that HEARING_INFO was created correctly. Explore
the NOPERCENT, NOCOL, and NOROW options for
this procedure.
*/
libname desktop
'C:\Users\Orysya\Desktop\SAS_Programming_II_Advanced_DATA_Step_Programmin
g\Assignment1_W1-4';
data ex_1 (keep=id hearing HEARING_INFO INCOME M_INCOME);
      set hearing;
      M_INCOME = Income/12;
      if missing(Hearing) then HEARING_INFO = 1;
            else HEARING_INFO = 0;
      label M_INCOME = 'monthly income';
      format M_INCOME FRACT9.;
run;
title 'Confirming Exercise 1.1';
proc freq data=ex_1;
      tables HEARING*HEARING_INFO;
run;

/*
Exercise 2.1. Using the IF-THEN/ELSE statement, create the following
three variables based on the variables in
the GRADE.SAS7BDAT data set:
1. MATH_POINT:
MATH_POINT is assigned to 4 if variable MATH = “A”
MATH_POINT is assigned to 3 if variable MATH = “B”
MATH_POINT is assigned to 2 if variable MATH = “C”
MATH_POINT is assigned to 1 if variable MATH = “D”
```

```
MATH_POINT is assigned to 0 if variable MATH = “F”
2. ENGLISH_POINT:
ENGLISH_POINT is assigned to 4 if variable ENGLISH ≥ 90
ENGLISH_POINT is assigned to 3 if variable 80 ≤ ENGLISH < 90
ENGLISH_POINT is assigned to 2 if variable 70 ≤ ENGLISH < 80
ENGLISH_POINT is assigned to 1 if variable 60 ≤ ENGLISH < 70
ENGLISH_POINT is assigned to 0 if variable ENGLISH < 60 and not missing
3. PE_GRADE:
PE_GRADE is assigned to “pass” if variable PE = 1
PE_GRADE is assigned to “no pass” if variable PE = 0
*/
libname desktop
'C:\Users\Orysya\Desktop\SAS_Programming_II_Advanced_DATA_Step_Programmin
g\Assignment1_W1-4';
data ex_2;
      set grade;
      if MATH = 'A' then MATH_POINT = 4;
      else if MATH = 'B' then MATH_POINT = 3;
      else if MATH = 'C' then MATH_POINT = 2;
      else if MATH = 'D' then MATH_POINT = 1;
      else if MATH = 'F' then MATH_POINT = 0;
      if ENGLISH >= 90 then ENGLISH_POINT = 4;
      else if 80 <= ENGLISH < 90 then ENGLISH_POINT = 3;
      else if 70 <= ENGLISH < 80 then ENGLISH_POINT = 2;
      else if 60 <= ENGLISH < 70 then ENGLISH_POINT = 1;
      else if ENGLISH < 60 AND not missing(ENGLISH) then ENGLISH_POINT =
0;
      if PE = 1 then PE_GRADE = 'pass';
      else if PE = 0 then PE_GRADE = 'no pass';
run;

/*
Exercise 3.1. Consider the following data set, PROB3_1.SAS7BDAT:
Notice that the SCORE variable contains missing values for some
observations. For this exercise, you need to modify the
SCORE variable. If SCORE is missing for the current observation, use the
SCORE value from the previous observation. The
resulting data will look as shown below:
*/
libname desktop
'C:\Users\Orysya\Desktop\SAS_Programming_II_Advanced_DATA_Step_Programmin
g\Assignment1_W1-4';
data ex_3;
      set prob3_1;
      retain score_;
      if not missing(score) then score_=score;
      drop score;
run;

/*
Exercise 4.1. Consider the following data set, PROB4_1.SAS7BDAT:
The SCORE variable is recorded at three different time points for each
subject. For this exercise, you need to modify
```

```
the SCORE variable. If SCORE is missing for the current observation, use
the SCORE value from the previous recorded
time point within each ID. The resulting data will look as shown below:
*/
libname desktop
'C:\Users\Orysya\Desktop\SAS_Programming_II_Advanced_DATA_Step_Programmin
g\Assignment1_W1-4';
data ex_4;
      set prob4_1;
      by ID;
      retain _score;
      if not missing(score) then _score=score;
      drop score;
run;

/*
Problem 1
The following data set (casecontrol.sas7bdat) contains hypothetical data
for a case control study. Cases
have values of 1 in the CASE CONTROL variable and controls have values of
2. Only cases have data
for stage and grade variables.

Based on this data set, you need to create two variables (GRADE NEW and
STAGE NEW) by using
BY group processing. These two variables are created by assigning the
stage and grade of each case to
their matched control so that each case-control pair is then essentially
matched on grade and stage. The
resulting data set looks like the one below:
*/
data Problem_1;
      set
'C:\Users\Orysya\Desktop\SAS_Programming_II_Advanced_DATA_Step_Programmin
g\Assignment1_W1-4\casecontrol.sas7bdat';
      by ID;
      retain grade_new;
      if not missing(grade) then grade_new = grade;
      retain stage_new;
      if not missing(stage) then stage_new = stage;
run;

/*
Problem 2
You will work with the q2.sas7bdat data set for this problem. Here are
the
```

The data set is sorted by ID already. Just to be safe, you should sort the data by ID first.

```
rst and last 5 observations:
Write a program to subset the dataset above by taking only the two most
recent SCORES for each subject.
The resulting data set should look like the one below:
*/
data Problem_2 (drop=count);
      set
'C:\Users\Orysya\Desktop\SAS_Programming_II_Advanced_DATA_Step_Programmin
g\Assignment1_W1-4\q2.sas7bdat';
      count + 1;
      by id DESCENDING date;
      if first.id then count=1;
      if count <=2 then output;
run;
```