Total Points: 30/30

```
/*
Name: Orysya Stus
Due Date: August 8 2016
*/

/*
Exercise 5.1. Suppose that you would like to invest $1,000 each year at
a bank. The investment earns 5% annual interest, compounded
monthly (that means the interest for each month will be 0.05/12).
Write a program using an explicit loop (or loops) to calculate your
balance for each month if you are investing for 2 years (that means
you deposit $1,000 at the beginning of each year).
*/
data ex_1;
      do year = 1 to 2;
            capital + 1000;
            do month = 1 to 12;
                  interest = capital*(0.05/12);
                  capital+interest;
                  output;
            end;
      end;
run;

/*
Exercise 6.1. Consider the following data set, PROB6_1.SAS7BDAT:
ID G1 G2 G3 S1 S2 S3
1 1 A A C 3 4 9
2 2 A B F 3 7 4
3 3 A C B 5 8 9
Transform PROB6_1 to multiple observations per subject by using
array processing like below:
ID TIME GRADE SCORE
1 1 1 A 3
2 1 2 A 4
3 1 3 C 9
4 2 1 A 3
5 2 2 B 7
6 2 3 F 4
7 3 1 A 5
8 3 2 C 8
9 3 3 B 9
*/
data ex_2 (drop=G1-G3 & S1-S3);
      set PROB6_1.SAS7BDAT;
      array g[3];
      do time = 1 to 3:
            grade = g[time];
            if not missing(grade) then output;
      end;
      array s[3];
      do time =1 to 3;
            score = s[time];
            if not missing(score) then output;
```

```
      end;
run;

/*
Problem 1
You are given the SAS data set SPEED (speed.sas), created by running the
program below. Create a new
data set SPEED2 from SPEED, with some new variables. The new variables
LX1 - LX5 are the natural
log of the variable X1 - X5, and variables SY1 - SY3 are the square roots
of the variables Y1 - Y3. Use
arrays to create the new variables.
data speed;
input X1-X5 Y1-Y3;
datalines;
1 2 3 4 5 6 7 8
11 22 33 44 55 66 77 88
;
*/
data SPEED2 (drop=X1-X5 Y1-Y3);
      set SPEED;
      array speedo X1 X2 X3 X4 X5;
      do over speedo;
            LX = log10(speedo);
            if not missing(LX) then output;
      end;
      array squared Y1 Y2 Y3;
      do over squared;
            SY = sqrt(squared);
            if not missing(SY) then output;
      end;
run;

/*
Problem 2
You will work with the dna.sas
```

le that reads 15 DNA sequences (See below). The length of each sequence is 60 characters. Based on these DNA sequences, create 60 variables, D1 - D60. D1 will hold the DNA at
the

```
rst position, D2 will hold the DNA at the second position, and so on. You
must use array processing
to complete this problem. Hint: use the SUBSTR function.
data dna;
length dna $ 60;
input dna $;
datalines;
TGGAAGGGCTAATTTGGTCCCAAAAAAGACAAGAGATCCTTGATCTGTGGATCTACCACA
TGATTGGCAGAACTACACACCAGGGCCAGGGATCAGATATCCACTGACCTTTGGATGGTG
CTTCAAGTTAGTACCAGTTGAACCAGAGCAAGTAGAAGAGGCCAAATAAGGAGAGAAGAA
CAGCTTGTTACACCCTATGAGCCAGCATGGGATGGAGGACCCGGAGGGAGAAGTATTAGT
GTGGAAGTTTGACAGCCTCCTAGCATTTCGTCACATGGCCCGAGAGCTGCATCCGGAGTA
CTACAAAGACTGCTGACATCGAGCTTTCTACAAGGGACTTTCCGCTGGGGACTTTCCAGG
GAGGTGTGGCCTGGGCGGGACTGGGGAGTGGCGAGCCCTCAGATGCTACATATAAGCAGC
TGCTTTTTGCCTGTACTGGGTCTCTCTGGTTAGACCAGATCTGAGCCTGGGAGCTCTCTG
GCTAACTAGGGAACCCACTGCTTAAGCCTCAATAAAGCTTGCCTTGAGTGCTCAAAGTAG
TGTGTGCCCGTCTGTTGTGTGACTCTGGTAACTAGAGATCCCTCAGACCCTTTTAGTCAG
TGTGGAAAATCTCTAGCAGTGGCGCCCGAACAGGGACTTGAAAGCGAAAGTAAAGCCAGA
GGAGATCTCTCGACGCAGGACTCGGCTTGCTGAAGCGCGCACGGCAAGAGGCGAGGGGCG
GCGACTGGTGAGTACGCCAAAAATTTTGACTAGCGGAGGCTAGAAGGAGAGAGATGGGTG
CGAGAGCGTCGGTATTAAGCGGGGGAGAATTAGATAAATGGGAAAAAATTCGGTTAAGGC
CAGGGGGAAAGAAACAATATAAACTAAAACATATAGTATGGGCAAGCAGGGAGCTAGAAC
;
*/
data Problem_2;
      do until (last.dna);
            set DNA;
            by dna;
            array D[60] D1-D60;
            do i = 1 to 60;
                  SUBSTR(dna, 1, 15);
            end;
      end;
run;
```