Homework#3 (Due : May 27th)

- Deadline : May 27th, 11:59pm
- Where to submit : eClass 과제방 (http://eclass.cau.ac.kr)
- How to submit :
  - Create a folder. The name of the folder should be "studentID#.hw4". (ex) 20111499.hw4
  - We have five problems in hw3. Make five C source files. The name of the source files should be the format "studentID#.hw4.#.c"
    - ex) 20111499.hw4.1.c , 20111499.hw4.2.c , 20111499.hw4.3.c 20111499.hw4.4.c , 20111499.hw4.5.c
  - In each source file .c, your code must have comments that include your name and student_id#
  - put the five source files into the folder we created.
  - Compress the folder into a zip file using WinZip (Don't use ALzip). The name of the zip file should be "student#.hw4.zip". (ex) 20111499.hw4.zip
  - Upload the zip file into eClass website.
  - If you don't follow above instructions, you may get penalty in your score.
  - You must do programming yourself.

Prob1. [Parlindrome] A parlindrome is a word or a sequence of words that has the property of reading the same in either direction. Examples of the parlindrome are "level", "radar", "hannah", and etc. Write a program that checks whether a given string is parlindrome or not. The program should get a string from a keyboard, then print whether the input string is parlindrome or not. If it is a parlindrome, print "parlindrome". If not, print "NOT parlindrome".

(example)
> prob1.exe
level
parlindrome

> prob1.exe
alice
NOT parlindrome

Prob2. [KIN] Korean Gymnastic Association is trying to modify the scoring system in order to improve its fairness. In the previous system, each of 5 judges gives a score from 1 to 10. The sum of the scores except the maximum and the minimum score becomes the final total score.

To improve the previous system, we first remove the maximum and minimum scores out of the 5 scores. Then, if the difference between the maximum and the minimum scores out of the remaining 3 scores is 4 or more, the scores need to be re-evaluated.

Write a program that takes 5 scores as an input and print the total score. If the score needs to be re-evaluated, print KIN (Keep In Negotiation).

```
Input Example :
10 8 5 7 9

Output Example:
24
```

```
Input Example :
10 3 5 9 10

Output Example:
KIN
```

Prob3.[Words Separation] Write a program that prints a list of words from a sequence of user input string lines. The program should get strings with several lines from keyboard input(standard input). To indicate the end of user input strings, the last line of the input should be a single character "!" followed by ENTER key input. After receiving the input strings, the program should print each word in the string lines one by one in order. You may assume the maximum possible length of a word is 30 and maximum possible number of input lines is 50. Also assume that a word consists of only alphabets. Do not use `strtok` function.

The example is as follows :

```
> prob3.exe
The Department of      Computer Engineering
offers students,the challenge   ,   of a dynamically
evolving science.
!

The
Department
of
Computer
Engineering
offers
students
the
challenge
of
a
dynamically
evolving
science
```

Prob4.[Standard Deviation] Write a program that gets an arbitrary number of positive float values until -1 is received from a keyboard (standard input) and print the mean and the standard deviation of the positive float values. Please use following formula for computing the standard deviation :

standard deviation $s_N = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$, $\bar{x}$: mean

```
#include <stdio.h>

int main()
{
        float float_val;
        // declare variables here

        while (1) {
                scanf("%f",&float_val);


                // put your code here


        }


        // put your code here

        return 0;
}
```

Output :

```
> a.exe
1.5
2.3
4.7
-1
mean = 2.833333 , standard dev = 1.359738
```

Prob5.[Maze] The grid of hashes(#) and dots(.) is a two-dimensional array representation of a maze. In the two-dimensional array, the hashes represent the walls of the maze and the dots represent squares in the possible paths through the maze. Moves can be made only to a location in the array that contains a dot.

There is a simple algorithm for walking through a maze that guarantees finding the exit (assuming that there is an exit). If there is not an exit, you will arrive at the starting location again. Place your right hand on the wall to your right and begin walking forward. Never remove your hand from the wall. If the maze turns to the right, you follow the wall to the right. As long as you do not remove your hand from the wall, eventually you will arrive at the exit of the maze. There may be a shorter path than the one you have taken, but you are guaranteed to get out of the maze if you follow the algorithm.

Write a program that receives input string representing 12-by-12 character maze and display the path from the starting location of the maze to the exit. In the input string, the starting location is marked as "S" and the exit is marked as "E". For the display of the path, print the maze with current location marked as "X" for each move.

(Example)

```
> prob5.exe
############
#...#......#
S.#.#.####.#
###.#....#.#
#....###.#.E
####.#.#.#.#
#..#.#.#.#.#
##.#.#.#.#.#
#........#.#
######.###.#
#......#...#
############

############
#...#......#
X.#.#.####.#
###.#....#.#
#....###.#.E
####.#.#.#.#
#..#.#.#.#.#
##.#.#.#.#.#
#........#.#
######.###.#
```

```
#......#...#
############

############
#...#......#
SX#.#.####.#
###.#....#.#
#....###.#.E
####.#.#.#.#
#..#.#.#.#.#
##.#.#.#.#.#
#........#.#
######.###.#
#......#...#
############

############
#X..#......#
S.#.#.####.#
###.#....#.#
#....###.#.E
####.#.#.#.#
#..#.#.#.#.#
##.#.#.#.#.#
#........#.#
######.###.#
#......#...#
############

############
#.X.#......#
S.#.#.####.#
###.#....#.#
#....###.#.E
####.#.#.#.#
#..#.#.#.#.#
##.#.#.#.#.#
#........#.#
######.###.#
#......#...#
############


......
```

```
############
#...#......#
S.#.#.####.#
###.#....#.#
#....###.#.X
####.#.#.#.#
#..#.#.#.#.#
##.#.#.#.#.#
#........#.#
######.###.#
#......#...#
############
```

Sample C code where you can start.

```
#include <stdio.h>
#define N 12

/*
    your function definitions here
 */

int main()
{
    int i;
    int maze[N][N+1];
    /*
           your local variables here
     */

    /* get 12X12 maze from keyboard */
    for (i=0;i<N;i++) {
           gets(maze);
    }

    /*
           your code here
     */

    return 0;
}
```