Problem Statement

You are asked to develop a FORTRAN program that will provide engineers and scientists with some basic statistical values. The program would be able to provide the following on a set of numbers stored in an input data file:

1. Count how many number
2. Calculate the average
3. Find the maximum number and how many times it appears.
4. Find the count and the numbers that are below the average
5. Sort the numbers in ascending order.
6. Add a number to the array
7. Print the numbers to the screen
8. Save the numbers to the input data file
9. Exit

Program Details

At the start, the program will read from the input data file. The file read by the program should only contain numbers with each number is on a separate line. After, the program will load all the numbers in that file into an array with a maximum size of 100 elements. Then, the program will **repetitively** display the following menu:

```
1. Count
2. Average
3. Maximum
4. Values below Average
5. Sort ascending
6. Add a number
7. Print
8. Save
9. Exit
```

Following table shows the action that will be performed when a specific option is selected.

Option	Action
1	Count how many numbers are there, and then print the count.
2	Calculate the average, and then print it.
3	Print the maximum number and how many times it appears.
4	Count how many numbers are *less than* the average, then print the numbers and the count.
5	Sort the numbers in the array in ascending order and print them to the screen.
6	Read a number from the user, and then add it to the array.
7	Print the numbers in the array to the screen.
8	Save the numbers in the array to the input data file.
9	Terminate.
other	An appropriate error message will be displayed.

Sample Demo

The developed program should behave in a way similar to the following sample run. Numbers that are bold and yellow represent the user inputs.

INPUT.DAT

```
89.0
58.0
85.0
90.0
93.0
97.0
97.0
64.0
87.0
73.0
```

```
WELCOME TO MY STATISTICS PROGRAM

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
1
THERE ARE 10 NUMBERS IN THE FILE

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
2
AVERAGE= 83.3

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
3
MINIMUM= 97.0    NUMBER OF TIMES (2)

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
4
VALUES BELOW AVERAGE ARE:
58.0    64.0    73.0
COUNT= 3

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
5
58.0  64.0  73.0  85.0  87.0  89.0  90.0  93.0  97.0  97.0

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
6
ENTER A NUMBER TO ADD
22.0
ADDED NUMBER= 22.0

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
7
THE NUMBERS ARE:
58.0  64.0  73.0  85.0  87.0  89.0  90.0  93.0  97.0  97.0  22.0

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
8
THE VALUES ARE SAVED IN THE FILE

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
10
WRONG CHOICE

1. COUNT
2. AVERAGE
3. MAXIMUM
4. VALUES BELOW AVERAGE
5. SORT ASCENDING
6. ADD A NUMBER
7. PRINT
8. SAVE
9. EXIT
ENTER YOUR CHOICE
9
GOODBYE
```

Development Guidelines

- At the top of the code file, include your name, id number, serial number, and section number.
- Start developing the overall structure of the program (menus, loops, and condition) before coding any function.
- At time of launching the project, you may not be able to do all the subprograms. So, prepare the subprograms you already know in order to gain time at the end.
- You must follow all the good programming styles and conventions taught in the lab (indentation, comments, good naming, modularization, etc.).

Submission Guidelines

- You are expected to submit a soft copy of your project through **Blackboard** in the due date. The soft copy will consist of:
- One FORTRAN file contains the project source code.
- A word document that contains the test cases you have done to your program.

Important Notes:

- You may use functions and subroutines when you think it is appropriate and it will make your program easier.
- Part of the grade will be assigned to the program readability and modularity. This includes variables' names, subprograms' names, indentation, and some comments. It also includes the proper use of subprograms.
- You program should be ready to be compiled and executed at submission time. You will lose part of the grade if you submit a program that does not compile correctly or does not run as expected.
- You should provide enough test cases to show that your program is running correctly.
- Every student has to work individually on the project. Cheating or copying from others will result in zero grades.
- The project has 5% of the lab and course grade.

Needed topics:

- Selection constructs
- Subprograms
- Repetition constructs
- Arrays
- File processing

You may not be able to finish the project at this time, because we did not cover all of these topics yet but, you can start working on it from now.

Grading Policy

The weight for each part of the project is as follows:

- Working program 70%
- Modularityi 10%
- Indentation 10%
- Naming style and comments[ii] 10%

Important Date

- The last time for submission through Blackboard is on Friday June 4, 2010, 11:59 PM.
- Any submission after this time will be rejected.

[i] Modularity means the use of subprograms whenever needed.
[ii] Comments demonstrate the purpose of variables, subprograms, blocks of statements. Comments, like hints, should be short but informative.