Programming Assignment #1

CS 6353 Section 1, Fall 2015

In this programming assignment, you will be coding in FORTRAN. The requirements for your submission are listed in sections 1 and 2 below. *Follow the requirements carefully and precisely – they will be thoroughly tested.*

Section 3 provides a few example outputs. Section 4 provides some reference material that you may find useful.

As a reminder, the way I grade will be by running your program against a large test suite, which includes many examples that are valid, as well as many samples that are invalid. You will be graded by the percentage of test cases you get correct. I recommend you test thoroughly against good output to make sure your basic functionality is right; also test thoroughly against malformed output.

Note that there is a separate document in the class web site for how to access the reference environment.

1. Program Specification

1. Your program must read numbers from standard input.
2. When a user inputs to the program, individual numbers are always to be followed by hitting 'enter'.
3. The first number shall be an integer that indicates how many additional numbers will be input (called 'items' for the rest of this spec).
4. For all items (the subsequent numbers input after the last one), you must be able to accept any number in standard decimal notation that the underlying FORTRAN implementation can deal with.

Formal input spec in BNF:

```
INPUT             ::= WS INTEGER WS '\n' NUM_LIST;
NUM_LIST          ::= WS NUMBER WS '\n' NUM_LIST | ;
INTEGER           ::= [0-9]+;
NUMBER            ::= OPTIONAL_SIGN INTEGER |
                      OPTIONAL_SIGN FLOAT;
OPTIONAL_SIGN     ::= '-' WS | '+' WS | ;
FLOAT             ::= NO_LEADING | WITH_LEADING;
NO_LEADING        ::= '.' INTEGER;
WITH_LEADING      ::= INTEGER '.' OPTIONAL_INTEGER;
OPTIONAL_INTEGER ::= INTEGER | ;
WS                ::= [ \t]+ | ;
```

5. Your program must not produce any output before all numbers are entered, unless there is an error in the input, in which case your program should exit after it outputs only the string: ERR
6. When input is finished, your program should output to standard output four lines of output (in the following order):
    a. The string 'Sum:' followed by the sum of the items that were entered, followed by a newline.
    b. The string 'Average:' followed by the median of the items that were entered, followed by a newline.
    c. The string 'Minimum:' followed by the smallest item that was entered, followed by a newline.
    d. The string 'Maximum:' followed by the largest item that was entered, followed by a newline.
7. For all numbers you output, you should always provide exactly two digits of precision to the right of the decimal point, and adequate precision to the left.
8. Output is case sensitive – my test program requires you to get the case right. If you miss this, you will most likely fail all test cases and end up with a zero.
9. After you output, your program must exit.

Formal output spec in BNF:

```
OUTPUT   ::= SUMLINE AVGLINE MINLINE MAXLINE;
SUMLINE  ::= WS 'Sum:'     WS OUTFLOAT WS '\n';
AVGLINE  ::= WS 'Average:' WS OUTFLOAT WS '\n';
MINLINE  ::= WS 'Minimum:' WS OUTFLOAT WS '\n';
MAXLINE  ::= WS 'Maximum:' WS OUTFLOAT WS '\n';
OUTFLOAT ::= [0-9]+ '.' [0-9] [0-9];
WS       ::= [ \t]+ | ;
```

2. Other Requirements

You will receive a 0 on this if any of these requirements are not met!

10. The program must be written entirely in FORTRAN77 (do not use features from later versions of FORTRAN).
11. The program must live in a single source file called **assignment1.f**, which I will compile using: `gfortran assignment1.f –o assignment1`
12. The program must be compilable with gfortran – I will not run any other FORTRAN compiler.
13. The program must compile and run in the reference environment. Even if it works on your desktop, if it doesn't work in the reference environment, you will get a 0.
14. You must submit your homework before 11:55pm on the due date.
15. You must submit the homework through the course website, unless otherwise pre-approved by the professor.
16. You must submit ONLY the single source file, **assignment1.f**
17. You may not give or receive any help from other people on this assignment.
18. You may use references on the Internet to teach yourself FORTRAN.
19. You may NOT use code from any other program, no matter who authored it.

3. Test Cases

Below are three sample test cases for you, which I will use in my testing. Typically, I use anywhere from 20-50 test cases, and will definitely use these three. I strongly recommend you create your own test harness and come up with a large number of test cases to help you get the best possible grade.

For test cases, what one would type on the command line is **BLACK**, input is in **GREEN**, and output is in **BLUE**.

Case 1

```
./assignment1
3
1
2
3
Sum: 6.00
Average: 2.00
Minimum: 1.00
Maximum: 3.00
```

Case 2

```
./assignment1
3
187
4
99
Sum: 290.00
Average: 96.67
Minimum: 4.00
Maximum: 187.00
```

Case 3

```
./assignment1
4
1
2
3
Text input
ERR
```

4. Resources

An excellent introduction to FORTRAN77 is available here:
http://www.star.le.ac.uk/~cgp/prof77.html

gfortran web page:

https://gcc.gnu.org/fortran/

Another FORTRAN77 tutorial:
http://web.stanford.edu/class/me200c/tutorial_77/