Preamble

You may develop your code anywhere, but you must ensure it runs correctly on a Linux distribution before submission.

Boundaries

*It is the function of vice to keep virtue within reasonable bounds.* – Samuel Butler

Your task is to read in three numbers and then compute and print some statistics: the minimum, the maximum, and the mean of the three numbers. Then, you are to read in a fourth number and print some information on how the fourth number (let’s call it $n$) relates to these statistics.

If the $n$ is less than the minimum of the first three or greater than or equal to the maximum, your program should print a message indicating that $n$ is out of bounds. If $n$ is not out of bounds, but less than the average of the minimum and the mean, your program should print out a message that $n$ is in the lower region. If the first two cases do not apply and $n$ is less than the average of the mean and the maximum, your program should print a message that $n$ is in the middle region. If none of the previous cases are true, then $n$ lies in the upper region and a message should be printed to that effect.

An Example

Here is a transcript of an interaction testing the out-of-bounds message:

```
$ python3 stats.py
Give me an integer: 2
Give me another: 10
And another: 6

The minimum is 2
The maximum is 10
The average is 6.00000

Give me the fourth integer: 1

The number 1 is out of bounds.
```

While the verbiage of the output does not have to match exactly, the spacing of the output does.
Coding Style

You should provide the following functions:

- a main function
- a function that computes and returns the minimum of three given numbers
- a function that computes and returns the maximum of three given numbers
- a function that computes and returns the mean of three given numbers
- a function that computes and returns the mean of two given numbers
- a function that computes and returns whether or not a given number is in between two given numbers (inclusive on the low end, exclusive on the high end). This function will return either a True (for in bounds) or False (for out of bounds). Functions that return a Boolean value are known as predicate functions.

You should not include logic anywhere in your program that duplicates logic implemented in one of the above functions. In other words, you should reuse functions as much as possible. HINT: your main will never explicitly use <, <=, >, or >=.

Test Cases

You are to construct five test cases, named test.oobl, test.oobh, test.lr, test.mr, and test.ur. The file test.oobl should contain the numbers

2
10
6
1

one number per line, with no blank lines.

To see if your test case is correct, run the command:

$ cat test.oobl | python3 stats.py

You should see output similar to:

Give me an integer: Give me another: And another:
The minimum is 2
The maximum is 10
The average is 6.000000
Give me the fourth integer:
The number 1 is out of bounds.

This method of running the program is called “piping in the input from a file.” When you actually do this, the prompts your program makes for information will all be strung together on a single line. Don’t worry about it; it’s a natural consequence of the way the program was run.
Now create the other test cases. The extension \texttt{oobl} stands for “out of bounds, lower”. From this, you should be able to figure out what test data goes into the remaining files.

The superior student will vary the first three numbers for each test case. In addition, the superior student will provide additional test cases that test the boundary conditions of this problem. In general, boundary conditions can be thought of as where two similar inputs to a program produce dramatically different outputs. For example, for the region \texttt{2..10}, 5 is in bounds, but so is 6. Thus 5 and 6 do not test a boundary condition. On the other hand, 9 is in bounds but 10 is not (due to the high end being exclusive). In this case, the inputs 9 and 10 do test a boundary.

**Grading**

You will receive a less than perfect score for the following infractions:

- missing test cases
- the test data in your test case files does not match the extension of the test file (\textit{e.g.} you put the wrong kind of test data in \texttt{test.mr})
- your program fails on any of your test cases (in this case, you will receive zero credit)
- your program computes the wrong result for any test case, supplied or otherwise
- your program does not follow the style guidelines.

**Solution Path**

One possible path to coding a solution for this problem would follow the following outline:

- Define a \texttt{main} function to read the input into variables. Print their values, just to test the code.
- Define a function to find and return the minimum of the three numbers. Have \texttt{main} call that function and print the result. Test your code again.
- Define a function to find and return the maximum of the three numbers. Have \texttt{main} call that function and print the result. Test your code again.
- Define a function to find and return the mean of the three numbers. Have \texttt{main} call that function and print the result. Test your code.
- Define a predicate function to determine if the fourth number is out of bounds. Have \texttt{main} call that function and if the result is \texttt{True}, print an “out of bounds” message. Otherwise print an “in bounds” message. Test your code.
- Replace the “in bounds” message with logic to handle another test case. Add the logic one test case at a time, testing that the logic works each time.

**Submission Instructions**

Change to the directory containing your assignment. Do an \texttt{ls} command to make sure you are in the correct place. You should see, at least, your \texttt{stats.py} and associated test files. Extra files are OK. Submit your program like this:
submit cs100 XXX project1

Remember to replace XXX with your section number.