CSCI 102 Placement Exam
Dept. of Computer Science
Viterbi School of Engineering

1 Introduction

You may use Java or C/C++ to solve this problem. When you login to Vocareum, be sure to start the assignment that is labelled with the language you intend to use.

Once you start Vocareum, you may download the file skeleton source code file and work on your own machine, or use Vocareum's integrated editor and terminal. If you work in your own machine it will be easiest to copy/paste your final code over the skeleton code in the Vocareum editor and then submit your work.

To compile your code, you can click the **Build** button in the upper right. Compilation messages/errors will appear in the terminal area in the lower part of the window. To run your program, click the **Run** button. Some messages may appear in the terminal area below **but there will be NO text prompt asking you to type in numbers**, so just type input in the terminal window and remember to end with a line that contains `0 0`. If your program doesn't halt correct, just type `Ctrl-C`. Finally, when you are done with your program, be sure to click the **Submit** button to ensure your attempt is registered.

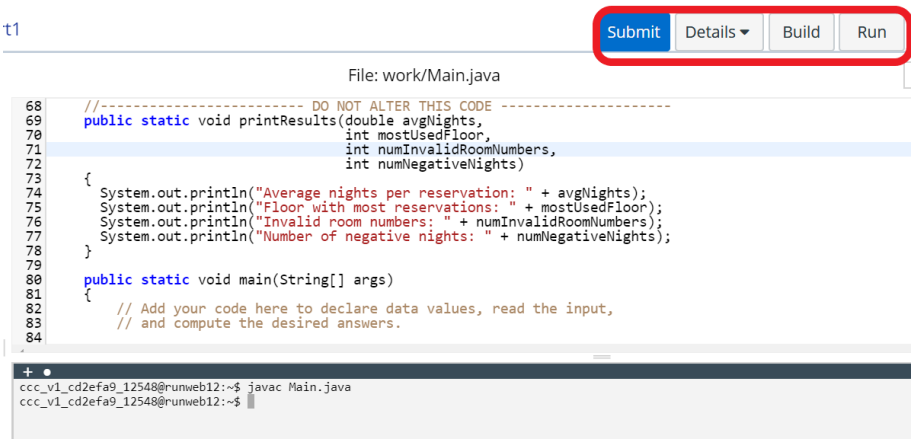

Figure 1: Vocareum Build, Run, Submit Buttons

2 The Problem

Given a hotel with 30 floors numbered 1-30 and up to 100 rooms per floor so that the lower 2 digits are the room number and upper digits are the floor number (e.g. room 1899 is the 99th room on the 18th floor), take reservation inputs that consist of pairs integers that represent the number of nights and a room number (each on a separate line) and terminated by `"0 0"`, and output:

(a) the average duration (number of nights) of a reservation

(b) the floor number (1-30) that had the most reservations. If there is a tie for the floor with the most reservations, you may output ANY one of the floors that tied.

For error checking of input values, you should ignore any input that has an illegal room number (i.e. above 3099 or below 100) or an input that has a negative number of nights for the reservation. Instead, count how many of these errors occur and output those values. So, in addition, to the outputting a. and b. described above, also output:

(c) the number of invalid room numbers entered

(d) the number of negative reservation durations.

We guarantee inputs from the user will contain at least one valid entry so that there will always be a floor with the most reservations and a non-zero average duration.

No other error conditions need be checked for and we guarantee the input will match our described format (two integers per line and terminated by a line with "0 0").

Your program should compute the desired values a. through d. described above. Then, We have provided a function, `printResults()` below that you MUST call to output those answers in a format that will our automated tests will recognized.

An example input and output sequence is shown below.

Sample input:

```
101 2
3099 3
2054 -1
3100 -1
99 2
-546 3
3064 2
0 0
```

In the above there are only 3 valid input lines (the first 2 and the last). There are 3 lines with invalid room numbers (3100, 99, -546) and 2 lines with negative night durations. Notice that one input line can contain both and invalid room number AND a negative night duration. 2.3333 is the average number of nights per reservation for the valid inputs. And floor 30 is the floor with the most reservations.

The correct output of your program for this input is shown below:

```
Average nights per reservation: 2.33333
Floor with most reservations: 30
Invalid room numbers: 3
Number of negative nights: 2
```

If you are unable to produce all 4 answers, you may pass dummy values (e.g. 0) to `printResults()`, but still produce and pass the answers you can correctly compute, so that you may still get partial credit.