

CSCI 151 Fall 2018

Homework 1

For this assignment, you will write a number guessing game. In this game, the player has multiple options, which will be presented in a menu:

- s: start a new game

A game consists of the player guessing numbers and being told whether the guess is too high or too low until the number is guessed correctly. At the end of a game, the user is told how many guesses it took for them to guess correctly

- q: quit

When the user quits the program, the average number of guesses made during the games is shown.

- n: set a new range

This allows the player to set a new maximum for the number range. The correct range should be shown in the following games. The default range is 0 to 9999.

- If an letter other than s, q, or n is entered, the player should get an error message and the menu should be presented again

In order to create secret numbers for the player to guess, your program will have to use a function called `rand`, which returns an integer between 0 and a very large integer (the maximum integer that can be stored on your computer). The value returned by `rand` will need to be constrained between 0 and the range used in your program. Every time `rand` is called, a new random number is returned.

```
int secret;  
secret = rand();
```

Computers cannot generate truly random numbers; the `rand()` function uses an algorithm to create numbers that are called *pseudorandom*. The algorithm starts with a number called a *seed*, and any particular seed produces the same series of pseudorandom numbers. The `rand` function starts with a default seed, which means that every time you use it, you get the same series of numbers. Luckily, there is a function called `srand` that allows you to *seed* the random number generator.

```
int seed;  
srand( seed );
```

If your program sets a different seed every time it is run, then each time it will generate a different series of random numbers. There are various ways to set a new seed every time your program is run. One way is to use the time, which is always changing. You may read on your own about how to use the function called **`time`**, which is defined in **`time.h`**.

When the program starts, the player should be welcomed and then presented with the menu. See the example interaction below. The player's entries are shown in red:

```
WELCOME to the guessing game!
MENU: (s) to start a game, (n) to set a new range, or (q) to quit:
t
UNRECOGNIZED command.
MENU: (s) to start a game, (n) to set a new range, or (q) to quit:
s
The secret number is BETWEEN 0 AND 9999. Guess:
4999
Too LOW! Guess:
7499
Too LOW! Guess:
8500
Correct: You TOOK 3 guesses!
MENU: (s) to start a game, (n) to set a new range, or (q) to quit:
n
Enter a new MAXIMUM:
999
MENU: (s) to start a game, (n) to set a new range, or (q) to quit:
s
The secret number is BETWEEN 0 AND 999. Guess:
50
Too HIGH! Guess:
25
Too HIGH! Guess:
24
Too HIGH! Guess:
23
Too HIGH! Guess:
22
Correct: You TOOK 5 guesses!
MENU: (s) to start a game, (n) to set a new range, or (q) to quit:
q
Thanks for playing. Your guess count AVERAGE 4.0
(program exits)
```

Because your program will be tested by another program, it is important that the program uses output messages that are machine-readable. Your output messages can vary, but they must include the phrases highlighted in bold above, **exactly as shown** (in caps, with the same spacing).

The programming assignment must be individual work. Do not discuss it with your classmates. Do not share your solution with anyone, in person or electronically, until the end of the semester. Do not discuss the assignment online. If you are having trouble, please ask one of your instructors or TAs.

Your program will be run through an automated plagiarism checker. If you are found to have cheated on any single programming assignment, **you will receive a 0 on ALL the programming assignments, past and future**, so consider this carefully. Late work on this assignment will be accepted, with a penalty of 20% per day.