Classwork 6: I Am Thinking of a Number.

In-class Date: Monday 11 March
Due Date: Monday 25 March *Monday after Spring Break*

Objectives

Practice writing a program that uses if statements and a while loop.

The Assignment

Write a program to play the game "I'm thinking of a number." The program will play the role of the person who has the "secret" number. Your program should prompt the user to guess a number. If user's guess is incorrect, your program should say whether the guess is too high or too low, and try again.

Example Compilation and Execution

```
[arsenaul@linux1 cw6]$ gcc -Wall thinking.c
[arsenaul@linux1 cw6]$ ./a.out
I'm thinking of a number between 1 and 100.
Guess my number.
Your guess? 13
Too low!
Your guess? 20
Too low!
Your guess? 35
Too low!
Your guess? 99
Too high!
Your guess? 74
Too high!
Your guess? 45
Too low!
Your guess? 84
Too high!
Your guess? 60
Too low!
Your guess? 70
Too high!
Your guess? 65
Too high!
Your guess? 63
Too low!
Your guess? 64
You got it!
[arsenaul@linux1 cw6]$
```

Starter Code

Use this code to help you get started. You must use a while loop in your program which terminates when the user guesses the correct number. Warn the user if they enter a number less than 1 or a number greater than 100. Notice how the starting point file uses `#define` to define a constant `SECRET_NUMBER` that holds the number your program is "thinking" of. That way, if you want/need to change the secret number, you only have to change it in one place.

```
/**************************************
** File:  thinking.c
** Author: <myName>
** Date:  <todaysDate>
** Section: CMSC104 Section 01
** E-mail: <myEmailAddress>
**
** This file contains the main program for <assignment>.
** <Explain what the assignment is asking you to do here.>
**************************************/

#include<stdio.h>
#include<stdlib.h>
#include<time.h>

// Constant value to use if you don't want to do any Extra Credit
#define SECRET_NUMBER 24

int main() {
   // Variable to use without any Extra Credit embellishments
   int guess;         /* user's guess */

   // Variable(s) to use for the Extra Credit embellishments
   int min;           /* lowest number of range to guess */
   int max;           /* highest number of range to guess */
   int secretNumber;  /* number to guess */

   // EC1: Seed the random number generator

   // EC2: Prompt for minimum value of range to guess

   // EC2: Prompt for maximum value of range to guess

   // EC2: Ensuring max > min

   // EC1 & EC2: Generate secret number within range

   // Prompt user to guess number
   // NOTE: Need to use min and max instead of 1 and 100 for EC
   // While loop until user guesses the secret number
   // NOTE: Need to use secretNumber instead of SECRET_NUMBER for EC
   // If user's guess is not in range, display "Guess must be between low and high!\n",
   // replacing low and high accordingly
   // else if user's guess is greater than secret number, display "Too high!\n"
   // else if user's guess is less than secret number, display "Too low!\n"

   // Prompt user for new guess

   // end of while loop

   // The user guessed the secret number, display "You got it!\n"

   return 0;
}
```

Extra Credit

Try to do the following embellishments:

Option 1: Instead of picking the same secret number each time, use a pseudo-random generator function. To do so, you must include the following header files:

```
#include <stdlib.h>
#include <time.h>
```

The first statement in your main function should set the random seed, as follows: `srandom(time(0));`

The time function returns the number of seconds since 00:00:00 UTC, January 1, 1970. So, each run of the program (started more than 1 second apart) will set then random seed to a different value.

Now each call to the function `random()` will return a pseudo-random number[1] as an int value. To convert this into a number that is within the range of allowed numbers, use the % mod operator: `n = random() % 100 + 1;` Then no matter what random returns, the variable n is assigned a number between 1 and 100.

Option 2: Ask the user for the range of numbers from which to choose the secret number (instead of always choosing a number between 1 and 100). This will affect how you implement Option 1.

Grading Rubric

- Typescript: 15 points
- Header comments: 2 points
- Body comments: 3 points

[1] `https://en.wikipedia.org/wiki/Pseudorandom_number_generator`

- Compiles: 30 points
- Proper logic: 50 points
- EC1: +5 points
- EC2: +5 points

What to Submit

Use the script command to record yourself compiling and running your program several times with different number guesses. (Do not record yourself editing your program!) Exit from script. Submit your program and the typescript file.

```
[arsenaul@linux1 cw6]$ submit cmsc104_arsenaul cw6 thinking.c typescript
```

Verify

Make sure you submitted the assignment correctly.

```
[arsenaul@linux1 cw6]$ submitls cmsc104_arsenaul cw6
```

Last modified: 08 February 2023