Assignment #1A (Extend-O-Credit eligible)

This assignment is designed for students who already know C (or a similar language) and want more of a challenge and want to work ahead (WAY ahead).

I mean it. This assignment is NOT for someone who is relatively new to programming.

It is expected that you will do this assignment without asking for help with programming in C. I am not willing to teach you material that you don't know in advance of when it's taught to everyone else (it's not fair to the other students who are doing the simpler assignment and have very little experience).

Simulate the game *Deal or No Deal* as a text console-based program. An online graphical version of this television game show can be found at http://www.mysteinbach.ca/game-zone/245/deal-or-no-deal/.

There are 26 "suitcases", each containing an amount of money (amounts listed in the online game). No two suitcases contain the same amount. Each game has the suitcase amounts assigned randomly. The contestant selects a suitcase number, which then becomes their suitcase. After picking the case, the contestant selects six of the remaining suitcases. The amounts contained within those cases are revealed, one at a time. After the sixth suitcase is revealed, The Banker makes an offer to buy the contestant's case for a certain amount, calculated by the program based on the cash amounts still in play. If the contestant accepts the deal, the game is over and the contestant's suitcase contents are revealed. If the contestant turns down the offer, they must then select another five of the remaining cases to reveal. The Banker proposes another deal, which is accepted or rejected. Play continues, revealing a lesser number of suitcases each time, until a deal is accepted or all suitcases are revealed.

The graphical and sound aspects of the online game should not be replicated. What is desired in this assignment is replication of the gameplay.

In determining a fair amount for The Banker to offer, you are allowed to average the remaining amounts and multiply that value by a random value between 75% and 110%. Alternatively, if you find a better way of doing it, you can use it, as long as it truly is "better" (yes, this is subjective). If you choose to use a formula that you found on the Internet, you absolutely **must** credit the webpage and/or author (if known) in a comment near where you implement it.

Helpful Hint: You will want to use srand(), rand(), fgets(), sscanf(), printf() / cout.

Again, it is assumed that you either know C well already or you can pick up what you don't know easily. It is **not reasonable** to expect me to teach you what you don't know in order to allow you to do this assignment.

You must use C (plus the addition of cout, cin, and the string class object) to complete this assignment. You are not allowed to use any other languages.

It is your responsibility to follow the course requirements as detailed in the course notes **and** the SET Coding Standards. Some of the items that apply to this assignment are (not limited to):

- Have appropriate header comments and descriptions.
- Use correct and consistent style.
- Use mixed case or underscores to separate words in identifier names.
- Always initialize your variables.
- Indent the bodies of decisions appropriately.
- Do not use global variables.
- Do not use scanf().
- Use function comments.
- Use prototypes.
- Do not use goto.

Call your project cA1a. Call your source code cA1a.c or cA1a.cpp to generate cA1a.exe. Do not create any other source files. Submit your ZIP file as described in the lecture on the SET Submission Standards into the "Assignment 1A (not 1)" assignment submission folder.

If you want to do easy user input of integers, you can use getNum(). It is a function that you will create for Assignment #2. It looks like this:

```
#pragma warning(disable: 4996)
int getNum(void)
{

/* function limitation: only accepts 120 characters of input */
   char record[121] = {0};
   int number = 0;
   /* NOTE to student:
    * indent and brace this function consistent with your other functions */

   /* use  fgets() to get a string from the keyboard */
   fgets(record,  121, stdin);
   /* extract  the number from the string; sscanf() returns a number
    * corresponding with the number of items it found in the string */
   if(  sscanf(record, "%d", &number) != 1 )
   {
        /*  if the user did not enter a number recognizable by
         * the system, set number to -1 */
        number  = -1;
   }
      return  number;
}
```

If you choose to use getNum(), you must put the above code in your source code or you will have a linker error.

If you do this assignment well, you may be eligible for a 50 mark bonus (out of 100)

and

your mark will also apply to Mini-Assignments 4 and 5 (you do not have to do them).

This only applies if you do the assignment well.

Conversely, if you do the assignment poorly, you will not get a chance to resubmit it.

It is vitally important to note that you give the user a good user experience in order to get a good mark.

If you hand in both Assignment 1 and 1A, Assignment 1A will not be marked.