

Lab 12: Random numbers, nested if-else statements, and switch statements**Due:** 20/24/24

This assignment consists of two parts in which you will solve the same problem but using two different control structures.

In the first part of the lab, you will practice decision making using if statements. In the second part of the lab you will use a switch statement to make the same decisions.

Problem: Suppose you are asked to simulate a Magic 8-Ball. The Magic 8-Ball is a fortune telling toy. The player asks a question and the Magic 8-Ball randomly answers the question with one of 20 possible answers. A description of the game, along with the twenty answers can be found on this [Wikipedia page](#). The Wikipedia page includes a link to an online version of the game.

For this program you only need to use the following 5 answers according to the following table:

Random Number	Answer
0	Yes
1	Maybe
2	No
3	Ask again later
None of the above	I don't know

The answers will be generated at random using the following expressions:

- 1) `srand(static_cast<int>(time(NULL)))` to seed the random number generator
- 2) `(rand() % (highRange - lowRange + 1)) + lowRange` to generate the random number

Your task: implement in C++ the algorithm solution shown below.

Algorithm solution (in pseudocode):

To solve this problem your program must perform the following tasks:

Part A: Solves the problem using **multi-branch if-else statements**

Declare the function prototypes for the functions above main()

Declare a variable named **randomNumber** that holds **whole** numbers

Declare a variable named **lowRange** that holds **whole** numbers and initializes it to 0

Declare a variable named **highRange** that holds **whole** numbers and initializes it to 4

Seed the random number generator using expression 1) shown above

Prompt the user to enter a question

Ignores the user input

Call function **randNumGen()** to generate a random number and assign it to **randomNumber**

Display title "Part A solution"

Display the message shown below

"Answer: ", call function **fortuneTellerA()** to get the answer

- 1) You **must** define function **randNumGen()** according to the following specifications:
 - I. The function receives two whole numbers: the first one is the upper boundary and the second one is the lower boundary used to generate a random number.
 - II. Returns the random number generated using expression 2)
- 2) You **must** define function **fortuneTellerA()** that uses **multi-branch if-else statements** to determine the answer to be returned based on the number received.
This function receives the random number (whole number) and returns the corresponding answer based on the table shown above.

Important: in this solution make your function directly return the answer in each branch of the multi-branch if-else statements.

Part B: Solves the problem using **switch statements**

- 3) You **must** define function **fortuneTellerB()** that uses a **switch statement** to determine the answer to be returned based on the number received.
This function receives the random number (whole number) and returns the corresponding answer based on the table shown above.

Important: in this solution declare a local variable that holds text and assign it the corresponding answer in each case of the switch statement. Upon exiting the switch statement, return the value that was saved in the local variable.

Once you have defined this function add the steps shown below at the bottom of your program and implement them.

Display title "Part B solution"

Display the message shown below

"Answer: ", call function **fortuneTellerB()** to get the answer

The program must compile without errors or warnings.

Open **lab12.cpp** in your IDE and implement the above algorithm (already provided in the source code as comments).

Note:

- Do NOT remove or modify the statements that I use to test certain things in your program.
- Pay attention to the input and the output formats. Your solution must behave exactly like mine.
- Carefully analyze the following figure and use it as a reference to ensure you do the right things.

```

Ask the Magic 8 Ball a question: Is it raining?

Part A solution

Answer: Yes

Part B solution

Answer: Yes

Testing your solution

```

- Test and compare your solution with mine for different inputs. Create your own **test plans** to test your program with the necessary inputs to ensure that **all** possible answers are generated.
- Your program must pass all my tests.
- You must define the most appropriate type of function, the parameter list (using the most appropriate parameters), and the body of the function to be implemented.

If you have concerns or specific questions, post them on the Discussion Board of Blackboard. Don't forget to include at the top of the program the comments shown below with your information (name, class and section number, etc.)

```

////////////////////////////////////
//
// Name: <Put your name here>
// Date: <Today's date>
// Class: <Your class number and section number, like: CSCI 1470.02>
// Semester: <This semester, like: Fall 2012>
// CSCI/CMPE 1470 Instructor: <Your lecture instructor's name>
//
// Program Description: Enter here your description of what the program does
//
////////////////////////////////////

```

When done, submit your solution through Blackboard using the “Assignments” tool. Do Not email it.

Paste the **link** to your final solution along with your **source code** in the textbox opened when you click on **Create Submission** before you click on **Submit**.

The following is the basic criteria to be used to grade this part of the assignment:

You start with 100 points and then lose points as you don't do something that is required.

-5: Minor mistakes

wrong identifiers

wrong variable types

no comments or too few comments in source code

(Part A) doesn't properly determine the answer corresponding to the number received (each random number)

(Part A) doesn't have a default case

(Part B) doesn't properly determine the answer corresponding to the number received (each random number)

(Part B) doesn't have a default case

incorrect input format

incorrect output format

-10: Moderate mistakes

(**Part A**) doesn't **directly return the answer** in each branch of the multi-branch if-else statements

(**Part B**) doesn't **use a local variable to hold the corresponding answer** for each case in the switch statement

incorrect type of the function (each)

incorrect function call (each)

incorrect type of parameters (value or reference)

-20: Major mistakes

missing or wrong implementation of function **randNumGen()**

program does not pass all tests

program does not implement the provided algorithm

Incorrect/missing source code

Incorrect/missing link to your solution

-40: (**Part A**) missing or wrong implementation (doesn't use **multi-branch if-else statements**) of function **fortuneTellerA()**

-40: (**Part B**) missing or wrong implementation (doesn't use **switch statements**) of function **fortuneTellerB()**

-50: program doesn't compile

-100: The code submitted is not your creation (you got it from a web site or another person)

-10: Late

Important: more points may be lost for other reasons not specified here.

Sample runs of my program:

```
Ask the Magic 8 Ball a question: How are you?
Part A solution
Answer: Maybe
Part B solution
Answer: Maybe
Testing your solution
```

```
Ask the Magic 8 Ball a question: What time is it?
```

```
Part A solution
```

```
Answer: I don't know
```

```
Part B solution
```

```
Answer: I don't know
```

```
Testing your solution
```

```
AAAsk the Magic 8 Ball a question: m I the greatest?
```

```
Part A solution
```

```
Answer: Ask again later
```

```
Part B solution
```

```
Answer: Ask again later
```

```
Testing your solution
```