Lab 11: Making decisions **Due:** 10/26/22

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

Part A: Solves the problem using **complex conditional expressions**

In this first part of the lab you will practice using if statements with complex conditions to perform decision making. First you will review a program that demonstrates the use of if statements with complex conditions, then you will write your own C++ code to practice this skill.

Example Program

Our example program simulates how a car insurance company determines your premium. The user provides the requested information and the program determines the insurance premium. The insurance premium is based on the user's gender, age, and number of traffic tickets he has received (numTix). The formula for determining the insurance premium is:

Gender	Age	Premium
Male	Under 21	1500 + 200 * numTix
Male	21 to 29	1200 + 100 * numTix
Male	Over 29	1000 + 100 * numTix
Female	Under 21	1200 + 200 * numTix
Female	21 and over	1000 + 100 * numTix

Open file **carInsDiagram.pdf** to see a decision-making graph for the car insurance premium.

The **carInsComplex.cpp** program provided illustrates the skills you are learning in this lab. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

Once you understood this program start working on the following problem.

Problem: Suppose you are asked to write a program that displays the corresponding telephone digit for a given letter. The user must enter the letter in upper case and a function returns the corresponding digit as a character. If an invalid value is entered (a lower-case letter or a nonalphabetical character) the program displays an error message.

The letters and digits on a telephone are grouped this way:

2 = ABC	4 = GHI	6 = MNO	8 = TUV
3 = DEF	5 = JKL	7 = PQRS	9 = WXYZ

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:

Declare a variable named letter to hold the letter entered by the user
Declare a variable named digit to hold the digit returned by the functions
Display the title "Converts a capital letter to a digit on the telephone"
Prompt the user to enter a single capital letter
Get the value from the keyboard and store it in the corresponding variable
Display "Solution A"
Call **letter2digitA()** to get the corresponding digit for the given letter and assign it digit
If there is a matching digit for the letter entered
Display the message "letter ", letter, " corresponds to digit ", digit, " on the telephone"
Otherwise
Display the message "there is no matching digit for the letter ", letter, " entered"

You **must** define a value-returning function named **letter2digitA()** that uses **complex conditional expressions**[1] in **if-else statements** to determine the letter to digit correlation.
This function receives a character. If the character corresponds to any capital letter of the English alphabet it returns the corresponding telephone digit as a character; otherwise, it returns character '?' to indicate that an invalid character has been entered.

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

Once you finished part A go over the example on the use of **multi-branch if statements** to write the car insurance premium program.
The **carInsNested.cpp** program provided illustrates the skills you are learning in this lab. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

Once you understood this program solve the letter to digit problem using this type of structure.

You **must** define a value-returning function named **letter2digitB()** that uses **multi-branch if-else statements**[2] to determine the letter to digit correlation.
This function receives a character. If the character corresponds to any capital letter of the English alphabet it returns the corresponding telephone digit as a character; otherwise, it returns character '?' to indicate that an invalid character has been entered.
Once you have defined this function add the steps shown below at the bottom of your program and implement them.

Display "Solution B"
Call **letter2digitB()** to get the corresponding digit for the given letter and assign it digit
If there is a matching digit for the letter entered
Display the message "letter ", letter, " corresponds to digit ", digit, " on the telephone"
Otherwise
Display the message "there is no matching digit for the letter ", letter, " entered"

---

[1] **Complex conditional expressions** require you to combine relational expressions **using** Boolean (logical) operators.
[2] **Multi-branch if-else statements** require you to nest the if-else statement in a way that gets the same results as using complex conditional expressions **without using** Boolean operators.

IMPORTANT:

In **both solutions** (**A** and **B**), to determine the corresponding digit for a letter, you **must** work with **range** of letters (for example, if the letter is within the range 'M'-'O', the corresponding digit is '6'). **Tip:**Review example **ifandex.cpp** posted on Wednesday 10/12, it will help you with both solutions.
The program must compile without errors or warnings.

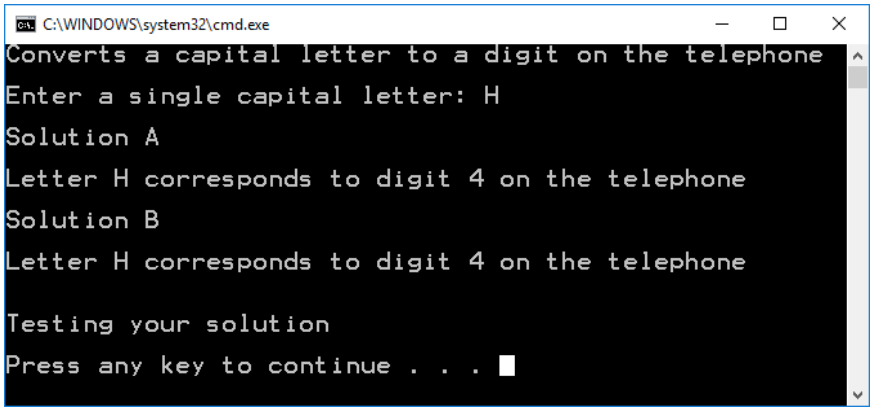
Open **lab11.cpp** in your IDE and implement the above algorithm (already provided in the source code as comments).

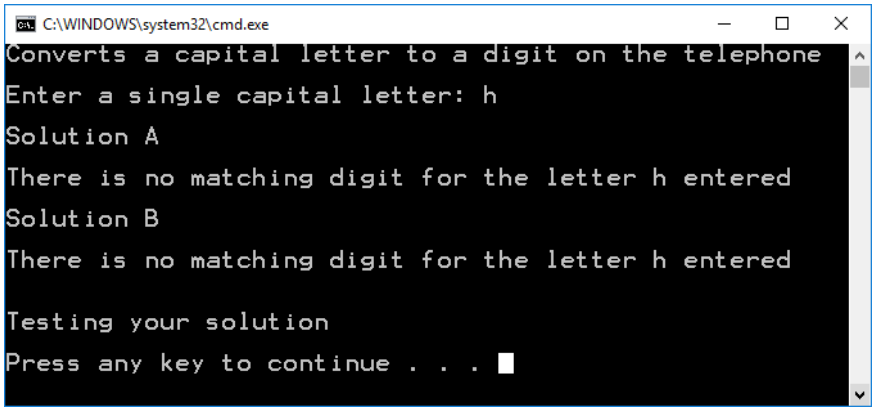
Note:

- Do NOT remove or modify the statements that I use to test certain things in your program.
- Run my sample solution to know how your program must behave. Pay attention to the input and the output formats. Your solution must behave exactly like mine.

  https://replit.com/@GDietrich/1370-lab11sample

- Carefully analyze the following figures and use them as a reference to ensure you do the right things.





- I am posting the executable of my solution for your reference. Test and compare your solution with mine for different letters and other characters. Create your own **test plans** to test your program with the necessary inputs to ensure that **all** possible letter to digit correlations are generated. Include tests for invalid inputs.
- Your program must pass all my tests.

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 1370.02>
// Semester: <This semester, like: Spring 2012>
// CSCI/CMPE 1370 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 solution and the source code in the textbox corresponding to Text Submission (click on the Write Submission button) 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.

1) -5: wrong identifiers (letter, digit)
2) -5: wrong variable types
3) -5: no comments or too few comments in source code
4) -5: doesn't display the output as specified in algorithm
5) -5: (Part A) doesn't properly determine letter to digit correlation (each group of letters)
6) -5: (Part A) doesn't take care of invalid inputs
7) -40: (Part A) missing or wrong implementation (doesn't use **complex conditional expressions)** of function **letter2digitA()**
8) -10: (Part A) does not work with range of letters
9) -5: (Part B) doesn't properly determine letter to digit correlation (each group of letters)
10) -5: (Part B) doesn't take care of invalid inputs
11) -40: (Part B) missing or wrong implementation (doesn't use **multi-branch if-else statements)** of function **letter2digitB()**
12) -10: (Part B) does not work with range of letters
13) -30: incorrect implementation of the function (each) [For example, wrong type of function]
14) -10: incorrect function call (each)
15) -10: incorrect type of parameters (value or reference)
16) -5: incorrect input format
17) -5: incorrect output format
18) -50: program doesn't compile
19) -20: does not pass all tests
20) -20: program does not implement the provided algorithm
21) -20: Incorrect/missing source code
22) -20: Incorrect/missing link to your Repl.it solution
23) -100: The code submitted is not your creation (you got it from a web site or another person)
24) -10: Late

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

The following are sample runs of my solution: