Name: _____

GEU111 Engineering Problem Solving with Computations
High-Tech Tools and Toys Lab
QUIZ 2 – A Five-Bit Decoder

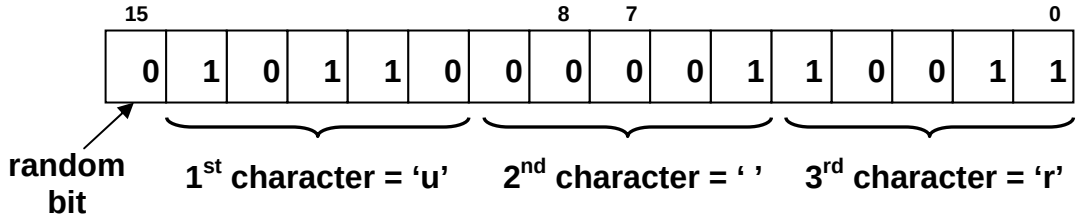
This quiz has three parts A-C. In a source file `Quiz2_xy.cpp` (where "xy" are your initials), write a `main` program that prompts the user to enter 1 to run Part A, 2 to run Part B, 3 to run Part C, and 99 to end. Write each part of the quiz in a void function as below. You will receive 25 Points for the correct header and a functioning "`main`" that prompts the user and call the respective functions.

You and a friend have become so concerned about warrantless wiretapping that you have decided to start communicating in code. This quiz involves writing a decoder for your cipher (here you assume the encoder is already written). For 10 points extra credit, write the five-bit encoder for Part B as well.

A. [25 Points] In a function `void Q_PartA(void)` implement an algorithm to decode the cipher mentioned above. In this part, assume the cipher works as follows: `0='.'` (period), `1=' '` (space), and `2-27` correspond to the letters `'a'` through `'z'` (your code doesn't allow capitalization or other punctuation!). Write a function `char Decoder(int n)` that returns a character corresponding to the integer as above. Note that you can do this in a considerably easier way than having 28 "`if else`" statements if you realize that in the ASCII representation `'b'` for example is equal to `'a'+1`. (You do not need to know the ASCII value of `'a'` to use this trick.)

Your `Q_PartA()` program should prompt the user to enter an integer between 0 and 27, use the `Decoder()` function to find the equivalent character, print out the decoded character, and then prompt the user to enter another integer. If the user enters a negative integer, the function should return to the main program.

B. [25 points] In a function `Q_PartB(void)` extend your decoder for a code that carries three characters per 16-bit integer by the following structure: bits 10-14 contain an integer between 0 and 28 that represent the first character, bits 5-9 contain an integer for the second character, and bits 0-4 for the third character. (Since it doesn't break on standard byte boundaries this code is considerably more secure than your code in Part A.)



Your `Q_PartB()` program should prompt the user to enter an integer, call a function `void Five_bit(int A, char& c1, char& c2, char& c3)` that uses a bitwise AND to extract the three component numbers from the input integer A, uses `Decoder()` to find the equivalent characters, and returns the three characters (`c1`, `c2`, and `c3`). `Q_PartB()` should output the characters to the screen. In the above example, the integer 22579 should output "`u r`".

C. [20 or 25 Points] In a function `Q_PartC(void)` implement an algorithm to read integers from a file that you can upload from Blackboard, decode the integers, and write out the clear text to the screen. (Note: `'endl'` should not be used here!) You have two options:

a. [25 Points] Use the `Five_bit()` function and the data file "`Quiz2_input.txt`".
b. [20 points] Else, use the `Decoder()` function with the data file "`Quiz2_easy_input.txt`".

Submit your `Quiz2_xy.cpp` file to Blackboard under the Quiz 2 assignment. Use the back of this sheet to write out a draft of your program before starting to type. Turn this sheet in as well as submitting your C++ source code to the Blackboard.