Program Purpose

Using Visual Studio 2017 create a console application. Create a program plan and then convert it into C++ statements. Practice debugging, declaring variables, type casting, formatting output, if statements and input /output to / from the console window and files, input validation, and loops.

Always bring to class

1. Gaddis' book, How-to handouts from the Blackboard and your class notes.
2. This assignment sheet & the grade sheet for this lab already printed out.
3. USB Flash drive(s) or other storage media.

LATE PROGRAMS will be not be accepted. Please be sure to upload compressed solution and Word document grade sheet to Canvas by due date/time.

Mandatory Instructions

Your next assignment will be to write a program to translate English to Pig Latin. The user will have an option to translate an input file, Program5Input.txt, or a single sentence entered at the console window.

Present the user with a menu at the console window like the one shown below. Remember to use do while loop for the menu. Validate user choice. Assume they will enter a number and not a string value like "hello." Display appropriate error message if the user enters anything other than 1 or 2.

```
            Menu:
================================
[1] to process a file
[2] to process a single sentence
================================
Your choice: 2_
```

User will chose [1] to process a file and [2] to process a single sentence. If the user selects a file option, then read the input file, Program5Input.txt, and translate all text in it (line by line). Place translated text into the output file, Program5Output.txt.

```
            Menu:
================================
[1] to process a file
[2] to process a single sentence
================================
Your choice: 1
File has been processed.

Press any key to continue . . .
```

If the user selects the single sentence option, then capture the input sentence and then show translated text in the console window.

```
            Menu:
================================
[1] to process a file
[2] to process a single sentence
================================
Your choice: 2
Your sentence:
Jack and Jill went up a hill to fetch a pail of water._
```

Rules for translating text to Pig Latin:

There are many dialects and forms of Pig Latin which vary from region to region, country to country, and language to language, as well as other similar, and dissimilar, Pig Latin-like 'languages'. The dialect shown here tends to hail from California/West Coast of the United States.

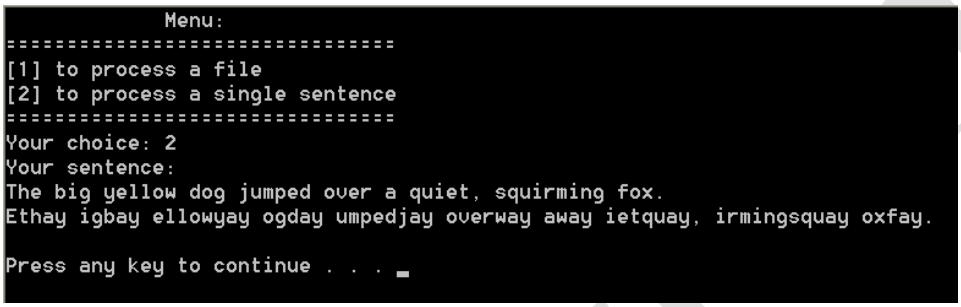
Basically, the Pig Latin system used here works as follows:

- Words that start with a vowel (A, E, I, O, U) simply have "way" appended to the end of the word
- Words that start with a consonant have all consonant letters up to the first vowel moved to the end of the word (as opposed to just the first consonant letter), and then "ay" is appended. NOTE: 'y' is considered a vowel in this context
- Ensure punctuation is preserved, i.e., periods and commas should appear after translated words in the same places
- Correctly translate "qu" (e.g., ietquay instead of uietqay) which means “qu” always stays together
- Differentiate between "y" as vowel and "y" as consonant (e.g. yellow = ellowyay and style = ylestay)
- Ensure proper capitalization (BONUS only)

Words may consist of alphabetic characters only (A-Z and a-z). There will be no contractions such as ‘ve ‘re ‘t ‘s etc. Sentence structure is preserved (i.e., commas, periods at the end and spacing). You may assume there will be **single blanks** between words and a single blank after commas and periods in paragraphs. The very last sentence will not have a space after the period.

BONUS

Ensure that word capitalization is preserved in the translated text.



Method:

Start by processing only user’s input from console, i.e., one word at first. Then build on that and add processing of a single sentence. Then move to modify your code to also process multiple sentences. Be careful not to simply copy-and-paste code, utilize same code, i.e., write functions that can be called in different circumstances.

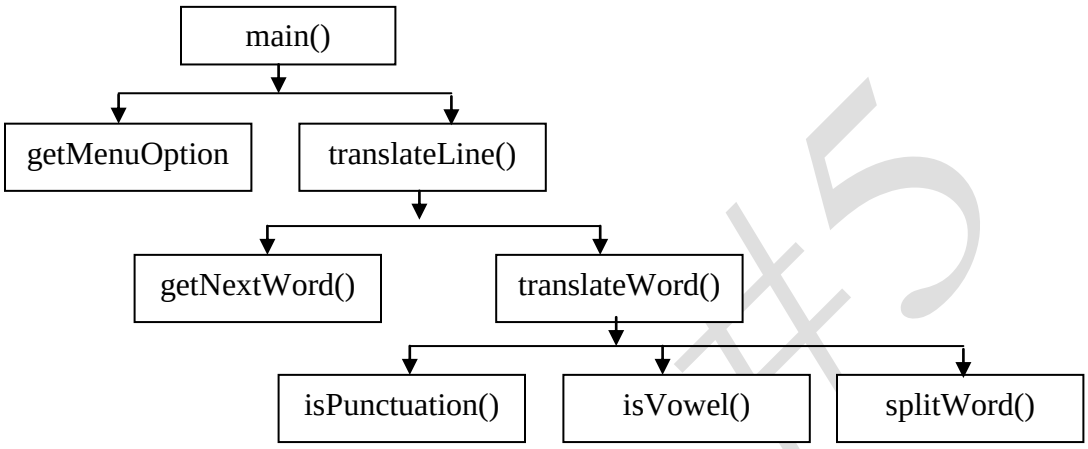
You have to utilize string processing similar to what we have done in class and in the last program, one character at a time. You may use function similar to isdigit() ➔ isalpha() when appropriate to check if character is alphabetic or not. You need to write other functions that will make the program easier to understand. Required functions are discusses in the next section.

START EARLY!!! Get help if you need it. This program is best not left undone until the night before it is due.

Program #5 DUE: Specified on Canvas

Structure chart:

This chart shows which functions call which other functions to subcontract some work. For example main() should only call getMenuOption and translateLine functions. These in turn call other functions as shown in the structure chart below.



Stepwise Refinement:

You have been given a ***suggested design*** in the structure chart which shows the functions that should be part of the program. Create first a "stub" program for your solution. A stub program should contain function prototypes, function calls and function headers – all with any needed parameters (arguments). Function bodies should be empty with correct return value and braces. The main function should declare variables needed, and should only contain function call to getMenuOption and translateLine. All of this should compile.

Function Prototypes

```
string getMenuOption();            // display menu and validate user input, return chosen option
string translateLine(string);      // process line one word at a time, building up the translated line
string translateWord(string);      // translate given word and return translated word
string getNextWord(string, int);   // pick off characters making up a word starting with given position
bool   isVowel(char);              // answers the question "is given character a vowel?"
bool   isPunctuation(char);        // answers the question "is given character punctuation?"
void   splitWord(string, int, string &, string &);// split given word starting at position given
                                                   // i.e., "hello" with position 1 would be split into two
                                                   // parts: "h" and "ello"
```

Program Documentation & Style:

1. Declare all variables and constants that your program uses at the beginning of your program.
2. Your program should include two types of comments:
   Header Comments at the top including lines with:
   - Your name, course name, and class time
   - Program assignment number, program file name (pgm5.cpp) and due date
   - A sentence or two explaining the purpose of the program
   - A description of the input data needed by the program when you run it
   - A description of the processing (calculations) done by the program
   - A description of the results (output) produced by the program

b. In-line comments: There should be an in-line comment for each main step in your program. In general, this means a comment with each group of C++ statements that handles the input, the processing and the output steps of your program.

3. Use meaningful identifier names
4. Include clear prompts for the user about entering the data.
5. Include clear descriptions of the results when you display them.

What to turn in?

1. Log into Canvas, locate this assignment, and upload the compressed project folders.
2. Upload the grade sheet (Word document) after you have edited it to provide requested information.
3. You're done.