Homework #10

This One's a Doozy

Assigned: November 27, 2017
Due: Dec. 7 by 11:59:59 PM

Until 2009, the US Postal Service printed a bar code on every envelope that represented the zip code using a format called POSTNET. We will be doing the same with only 5-digit zip codes. POSTNET consists of long and short lines, as seen below:

The POSTNET representation of 67260, WSU's zip code

In the program, the zipcode will be represented by an integer and the corresponding barcode will be represented by strings of digits. The digit 1 will represent the long bar, and the digit 0 will represent the short bar.
The first and last digits of a POSTNET code are **always** 1. Stripping these leaves 25 digits, which can be split into groups of 5. The above example translates into the following string and groups of five:

```
1011001000100101011001100 01
01100 10001 00101 01100 11000
```

Now, we look at each group of 5. There will **always** be two 1's. Depending on its location within the group, each 1 represents a number. When the numbers that the 1's represent are added together, you get that digit of the zip code. The table below translates the first group, which represents the number 6.

POSTNET Digits	0	1	1	0	0
Value	7	4	2	1	0

We see that the 1's correspond to the values of 4 and 2, respectively. Adding them up gives us 6, which is the first digit of the zip code (and also the fourth since the same group appears again, due to the zip code having two 6's)

In order to represent the number 0, the 1's will add up to a value of 11. This is done because of the requirement that every group of five always has two 1's in it.

Requirements:

- *NOTE*: Typically, I have tried to make the extra credit something that can be tacked on to the regular assignment. This is not the case here. The extra credit option will still use much of the same logic, but it will be designed in a fundamentally different manner
- Name your source file `program10.cpp`
- These are the first ~50 lines of code (Don't forget the required comment block!):

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

struct Zipcode {
    int romanZipcode;
    string postnetCode;    // Does not store the leading and trailing 1
};

// Accepts both roman and barcode formats, fills both struct members
Zipcode fillZipcode(const string zip);

// Conversion of roman zip code to bar code
string romanToPOSTNET(const int r);

// Conversion of bar code to roman zip code
int postnetToRoman(const string p);

void printRomanZip(const Zipcode zip);
void printPOSTNET(const Zipcode zip);

// Filename will be the roman zip, contents are graphical bar code
void writeToFile(const Zipcode zip);

/*
 * Argument is used to display appropriate prompt to user
 * Gets zip code from user
 * Creates and assigns to a Zipcode struct
 * Prints roman and graphical bar code to screen
 * Creates a file whose name is in the format ROMAN.txt where
 *     ROMAN is the roman zip code, and the contents are the
 *     graphical representation of the bar code
 */
void processZip(int prompt);
```

```
37
38  int main()
39  {
40      int mainMenu;
41
42      cout << "This program is able to convert zip codes to a POSTNET format "
43          << "and vice versa\n"
44          << "\t1. Convert zip code to POSTNET\n"
45          << "\t2. Convert POSTNET to zip code\n"
46          << "\t3. Quit\n";
47
48      do {
49          cout << "Please make your selection: ";
50          cin >> mainMenu;
51
52          switch(mainMenu) {
53          case 1:
54          case 2:
55              processZip(mainMenu);
56              break;
57          default:
58              if (mainMenu != 3)
59                  cout << "Invalid choice...\n";
60              else
61                  cout << "\n";
62          }
63      } while (mainMenu != 3);
64
65      return 0;
66  }
```

- This program will be compiled using a modified command
  - `g++ -Wall -std=c++11 -o [OUTPUT] program10.cpp`
- Implement and use all functions as given
- The `main()` function will **not** be altered
- You may declare and use additional functions if needed
- You may assume that a valid zip code is entered, in either format
  - You may also assume that no zip code will be provided that begins with a 0, even though they are legal
- When entering a bar code zip code, do include the leading and trailing 1

  - When storing the bar code string, however, the leading and trailing 1 are not to be included

- A sample run with zip code input will look something like this:

```
$ ./prog10
This program is able to convert zip codes to a POSTNET format and vice versa
        1. Convert zip code to POSTNET
        2. Convert POSTNET to zip code
        3. Quit
Please make your selection: 1

Enter a zip code in roman format (#####): 67260

Your zip code is 67260, and the bar code looks like this:

|   | ||     |       |     |   |   | ||     | ||       |
| | | | | | | | | | | | | | | | | | | | | | | | | | |

Your zip code was saved in the file 67260.txt
$
```

- A sample run with bar code input will look something like this:

```
$ ./prog10
This program is able to convert zip codes to a POSTNET format and vice versa
        1. Convert zip code to POSTNET
        2. Convert POSTNET to zip code
        3. Quit
Please make your selection: 2

Enter a zip code in bar code format (1's and 0's): 10110010001001010110011000 1

Your zip code is 67260, and the bar code looks like this:

|   | ||     |       |     |   |   | ||     | ||       |
| | | | | | | | | | | | | | | | | | | | | | | | | | |

Your zip code was saved in the file 67260.txt
$
```

- Yes, choices 1 and 2 provide the exact same output

  - The only difference is the prompt that the user sees, and what the user enters

Extra Credit:

- To score up to 125% (an extra 10 points) on this assignment, utilize a class instead

- It will have two constructors
  - The first will take an integer for the zip code
  - The second will take a string, the value of which is the barcode
- The class will only store the zip code in ONE of the formats (integer zip code or bar code string)
- The class will have three public member functions
  - A function to return the zip code as an integer
  - A function to return the zip code as a bar code string
  - A function to print the bar code to the screen OR to a file
    - One of parameters to the function will determine the print destination
- Naturally, you will be defining a class instead of a `struct`, so the beginning of your source file will not look the same
  - **HOWEVER**, the `main()` function will still look exactly the same
- You may declare extra helper functions as needed, but they *must* be privately declared

Hints:

- Making good use of the `string` class will be very important in this assignment
  - Namely, there are functions called `stoi()` and `to_string()` that should come in handy
- This assignment is a culmination of everything that has been learned up to this point
- DO NOT procrastinate
- The first line of the graphical representation is the only one that requires extra thought

Reminders:

- Be sure that your program includes your name, ID, description, etc. as shown in the General Homework Requirements Handout
- Use good style including indentation, comments, etc. Part of the grade will be for style and quality.
- Carefully test your program.
- You are welcome to write your program at home. If you do, be sure to compile and test it in the lab before submitting it.

How to submit your program:

- Submit the file `program10.cpp` electronically using `~cs211a/bin/handin 10 program10.cpp`