CSCI 2226 Due February 11

Program 2: Creating a Cryptogram

1 Goals for P2

1. To write a program that includes a main program, a class defined by the student.
2. To use and understand a template class and a standard C++ algorithm.
3. To understand how a`vector` object grows and how to use it effectively.

Before we can solve a cryptogram, we must be able to read it in from a file and store it. In the old days, we would use a character array of some fixed size and hope that the data in the input file was short enough to fit into that array. A more modern approach is to use an array that can be reallocated to be as long as necessary. C++ has a standard template class named `vector` that serves this purpose, and we will use `vector` in this project. By using code that you can see, I hope you will be able to understand what a template is and how an array can "grow".

2 Instructions.

2.1 Create the Cryptogram Class

- Define a C++ class named Crypto. This week the class will have just three data members, all objects of type `vector<char>`. The first will store the plaintext puzzle read in from a file. The second will be the encoded input message. The third will be the partially solved puzzle. More members will be added in P3.
- The Crypto constructor must take the name of an plaintext input file (type string) as its first parameter and a pointer to a Code object as its second parameter. Open the file and check for proper opening. Call fatal() and print an informative error message if the file does not open.

  Then read one line of the file into your plaintext vector. Use string::getline(). Add a null terminator to the end of the line. This will become a puzzle to solve. Finally, process the plaintext one character at a time until the null terminator is found:

  - If the plaintext character is NOT a letter a..z or A..Z, push it into both the encoded message and the solution.
  - If the plaintext character IS a letter, convert it to upper case. Then look up the corresponding letter in the Code and push the coded letter into the encoded message. Push a corresponding underscore into the solution vector.

- Define a function to print the encoded message, then print the contents of your solution vector and finally a blank line.

2.2 The main program and testing.

- In main, create a Code object and generate a code.
- In main, create a Crypto object and print the message.
- Use the file "p2.txt" as data. This will make it easy for both of us to check your results.
- Hand in source code for main, the Code and Cryptogram class, and your output.
- Grading criteria will be similar to those for P1 (6 points total). Basically, they ask you to follow instructions, produce readable output, and use safe programming techniques and reasonable OO style.
- As soon as you finish this project, start on P3.

2.3 Grading Criteria

Points	Criteria (Total of 6 points)
1	Has a main program that follows instructions.
1	Follows directions in implementing the Crypto class, including the use of vectors.
1	Defines class Crypto with an .hpp file and a .cpp file.
1	Opens an input stream and checks for proper opening.
1	Reads, stores, and encodes the input according to instructions.
1	Prints the encoded message and the solution line in a neat and readable way.