In this assignment, you will be writing a program in C++ to decode "puzzles" stored in a file. This assignment also covers C++ I/O, strings, vectors, and string streams.

Program Features

Your program will decode puzzles stored in a user specified file. The program should start by prompting the user to enter a file name. The file can contain **an arbitrary (0 or more) number** of puzzles. Decoding each puzzle should result in ASCII text, often a single sentence or paragraph, which can be represented using a C++ string. Once each puzzle has been decoded, the program will print a simple report. If the file is empty no additional information should be printed.

Each puzzle in the file is made up multiple lines of text. The first line is an integer value that specifies the number of words in the puzzle. After this integer, there is a line of text **for each word** in the puzzle, so if the integer on the first line of the puzzle were 2, there would be 2 additional lines each representing a single word. Each "word" line should be a list of integers. Once all of the words have been specified, each puzzle is separated by a blank line.

Here is a sample input file containing two encoded puzzles:

```
2
67 5 100 1 11 97 98 10 1 110
15 72 10 101 47 67 88 20 94 6 22 11

4
61 11 93 4 73 39 78 34 17 104
23 43 11 93 65 52 20 96 66 31 86 24 40 61 102 13 50 51
73 43 28 73 8 89 31 68 77 27 24 77 42 72 15 24 64 51
25 75 7 90 10 111 17 16
```

In the example above, the first puzzle has 2 words, so there are 2 lines of integers. Each **letter** in the word (a line) is represented by 2 integers. To obtain the corresponding letter you simply **add** the integers and **cast to char**. So 67 + 5 = 72, when converted to a char is the character 'H'. Each pair of integers is used only once, so you can then move on to the next 2 numbers 100 and 1. For reference you may want to look at the ASCII table.

Here is a sample run of the program:

```
Enter filename: puzzles.txt
Decoded messages:
Happy Bhutanese teacher's day!
Hello World!
```

Note that while "Hello World" is actually the first puzzle in the input file, the ordering in the output has changed. The results have been sorted using the `sort` function. So you should decode each puzzle, store the result in a `vector`, and then use the `sort` function before printing.

Make sure the prompts are identical to the example above; also notice the formatting for the output.

Hints

For this assignment, string streams are probably the simplest way to process each line of integers. When used with the extraction operator (>>), a string stream will behave the same way it does with an `ifstream` variable, so string streams can be used break up a line of text containing words or integers.

The example below would break a text file into "words" (chunks of text) and print each word on a separate line:

```
ifstream fin("story.txt");
string line;

while(getline(fin, line)
{
     // Make the line into a string stream.
     istringstream split(line);

     // Break a line down into "words".
     while (split >> word)
     {
          cout << word << endl;
     }
}
```

What to Submit

For this assignment you should submit your `main.cpp` file from inside of your project directory.

This assignment will be graded automatically. Test your programs thoroughly before submitting them. Make sure that your programs produce correct results for every logically valid test case you can think of. Do not waste submissions on untested code, or on code that does not compile with the supplied code from the course website.

Web-CAT will assign a score based on runtime testing of your submission; your best score will be counted; the TAs will later verify that your best submission meets the stated restrictions, and assess penalties if not.

To submit this assignment:

1. Visit http://web-cat.cs.vt.edu in your web browser.
2. Enter your Virginia Tech PID and password in the appropriate fields on the log-in screen, and make sure that **Virginia Tech** is selected as the institution. Click **Login**.
3. The Web-CAT home screen will display useful announcements and assignments that are currently accepting submissions. Find the assignment that you want to submit in the table, and click the "Submit" button next to it.

4. Click the **Browse...** button and select the file you want to upload. The homework assignments and programming projects for this course should be self-contained in a single **main.cpp** file, so you can simply select that one file.
5. Click the **Upload Submission** button. The next page will ask you to review your selection to ensure that you have chosen the right file. If everything looks correct, click **Confirm**.

The next page will show that your assignment is currently queued for grading, with an estimated wait time. This page will refresh itself automatically, and when grading is complete you will be taken to a page with your results.

Pledge

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the submitted file:

```
//    On my honor:
//
//    - I have not discussed the C++ language code in my program with
//      anyone other than my instructor or the teaching assistants
//      assigned to this course.
//
//    - I have not used C++ language code obtained from another student,
//      or any other unauthorized source, either modified or unmodified.
//
//    - If any C++ language code or documentation used in my program
//      was obtained from an allowed source, such as a text book or course
//      notes, that has been clearly noted with a proper citation in
//      the comments of my program.
//
//    <Student Name>
```

Failure to include this pledge in a submission will result in the submission being disallowed during code review.