Practice with file reading and prep for first test

FOOL-PROOF WAY TO CREATE A PROGRAM THAT READS FROM A FILE:

- Create a new project. The directory for this project **should not** be on a network drive; the path (location) **must** start with a drive letter. If in doubt, make the project on the Desktop.

- Make a new C++ file as normal. Edit code as normal.

- When you are ready to create a text file that this program will read from, right-click on "Source Files" in the left pane (Source Explorer) of Visual Studio. Choose Add -> New Item…

- From the left column, under Visual C++, choose "Utility," then select "Text File (.txt)" from the main area of the window. The "location" at the bottom of the window should be automatically filled in with the proper folder name. Choose a name for your text file (probably not "Text.txt") and choose "Add."

- Your text file is now created and you can access it from a tab in Visual Studio or from the left pane under Source Files.

Today's problems (skip the parts you have already done)

1. Create a program that reads in ints from a text file and prints their sum. Here is code to get you started:

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

using namespace std;

int main()
{
        ifstream infile("numbers.txt");
        int x;
        while (infile >> x)
        {
                cout << "I just read " << x << endl;
        }
        infile.close();
        return 0;
}
```

2. Create a program that reads in pairs of ints from a text file (interpreted as points in the x-y plane) and prints the cumulative distance between the points. Use the Day 11 file `structs.cpp` (see website) for a point struct and for a distance function. You should use the sliding window technique so that you will have access to the "current" and "previous" points inside the while loop that your program uses to read the code. Reading all the points into a vector will work, but is not the best way to solve this problem (because there's no point keeping the entire vector around when all you need is each pair of consecutive points).

3. Create a program that reads dates from a file, in the format "mm/dd/yyyy." Create a text file with a bunch of dates in this format, one per line. Start with the Day 12 code and add a function that opens this file and reads the dates into a vector (i.e., a vector<date>).

4. Continue with the problems you didn't get to from Day 12 (Feb 5).