EEN118 LAB NINE

This lab involves some scientific data processing. You will download a Geographical Database File which contains the coordinates of the boundaries of the 48 connected states and the five great lakes, and write a program that draws maps requested by the user.

Download the file "`usa.map`" from the class web site. The format of the file is very simple; it describes the outlines of the 48 older states plus the five great lakes. Here is the beginning of the file:

```
WA
122750   49000
117000   49000
117000   46333
116866   46000
119000   46000
122666   45500
122916   46166
123916   46250
124166   47000
124666   48333
122833   48166
122666   47416
122583   47333
122416   47333
122750   49000
-1 -1
ID
117000   49000
116000   49000
116000   48000
114500   46500
...
```

and it continues like that for a long time. The first line "`WA`" indicates that this is the description of a state: WA is the postal abbreviation for Washington. The next 15 lines give the coordinates of a point along the border of the state. The coordinates are actually longitude and latitude measured in thousandths of a degree, but you can treat them simply as `x` and `y` values. The two `-1`'s after the list of numbers are simply to give you an easy way of telling that the list has finished (no real data in this file is ever negative). Then you see "ID" introducing the next state, Idaho's, description.

The whole file is just like that. First a state's abbreviation, then a list of coordinates, then -1 -1, all repeated 55 times. Notice that the coordinates of the last point are the same as the coordinates of the first point (122750, 49000). This is true of every state's description; they all make nice closed figures. The coordinates 122750, 49000 represent the point 122¾ degrees North of the equator and 49 degrees West of the Greenwich meridian. It is where the U.S.-Canadian border meets the Pacific ocean.

At the end of the file, which looks like this...

```
WV
 82666   38333
 82000   37500
 81500   37250
 80250   37500
 79666   38666
 79333   38500
 78333   39500
 77833   39166
 77750   39333
 78166   39583
 78833   39500
 79500   39333
 79500   39666
 80500   39666
 80500   40666
 80666   40666
 81000   39500
 82666   38333
 -1 -1
 END
```

... you'll find the word END. The last state, West Virginia's description ends as usual with -1 -1, then the word END appears. No state has "END" as its postal abbreviation. There is no data following the "END".

For your information, these are the extreme data values that appear in the file:

Minimum Longitude (x)	70750
Maximum Longitude (x)	124666
Minimum Latitude (y)	25166
Maximum Latitude (y)	49333

The states all have their standard two-letter postal abbreviations. The great lakes are given three-letter abbreviations: LKE, LKH, LKM, LKO, and LKS.

1. Draw Washington

Write a program that opens a reasonably large graphics window (you decide the size), and draws the outline of the first state that appears in the data file (i.e. Washington). You will have to scale and shift the coordinates before plotting, as one of the points in Washington is (117000, 49000) and there's no way you're going to get a window that big. Make sure your picture comes out the right way round:

2. Draw any State

Modify your program so that it asks the user to enter a state's (or lake's) abbreviation, and then draws that state. It should not draw anything else, just the outline of the selected state. Do not worry about sizing the window to fit that state properly. Make the window big enough to draw the whole country, and just draw the one state in its correct position.

3. Did you get it Right?

Two of the states, Michigan (MI) and Virginia (VA) are not contiguous; they come in two sections separated from each other by water. The two sections of these states have their own descriptions in the data file: there are two sections beginning with VA and two beginning with MI. Make sure that if the user requests a non-contiguous state, all of its portions are drawn.

4. Make it Incremental.

Add a loop to your program. After creating the window big enough to draw the whole country, it should enter the loop, repeatedly asking the user to enter a state abbreviation, and adding the outline of that state to what has already been drawn, so that the user can build up a map of many states if desired. Remember to open the file each time round the loop, and close it before the end of the loop; that way you'll be able to read it afresh from the beginning each time.

5. Finally make it able to do Everything.

If the user enters ALL as the desired state abbreviation, your program should draw the outlines of all the states (in the same window), so that a map of the whole country appears. Of course the program should still draw individual states if the input is not "ALL".

6. ***(Extra Credit)*** Make it into an Interactive Coordinate Finder.

      There is a function in the library called `WaitForMousePress`. It takes two parameters, which must be `int` *variables*. When called, the function waits until the user (you) clicks the mouse somewhere within the graphics window. It then returns, setting the two parameters to the coordinates of the mouse press. This is how you might use the function in a graphics program:

```
int x, y;
WaitForMousePress(x, y);
cout << "Mouse pressed at " << x << "," << y <<"\n";
```

Modify your program so that if the user types `Q` (for Query) instead of a state abbreviation, it allows him or her to click the mouse somewhere on the map, and prints out in a reasonably neat way, the geographical coordinates of the point indicated. That is, you should convert the values of x and y back from pixel coordinates to Longitude and Latitude. If the user clicks somewhere around Miami, the output should be something in the region of:
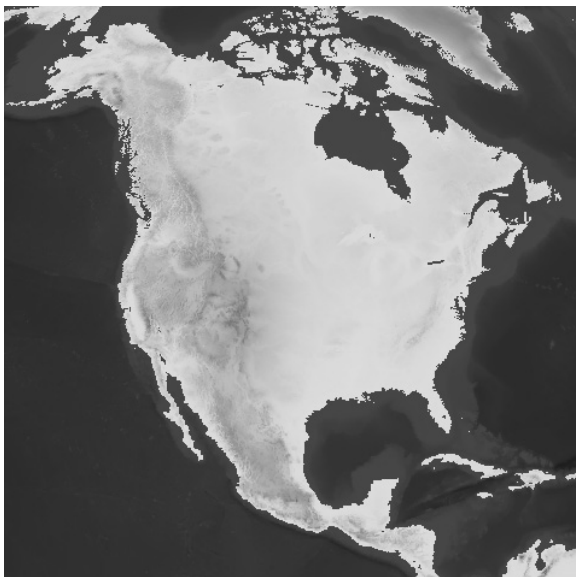
<div align="center">

`80.4 W, 25.8 N`

</div>

7. ***(Extra Credit)*** Colour it in.

      If the user enters "MAP", your program should behave exactly as it would if "ALL" had been entered - it should draw the whole country. Except that it should also colour in the interior of each state in a different colour. You don't have to give each state a truly different colour, I don't want you wasting time working out 53 different colours; make it like a real map: use five or six different colours perhaps picking one at random for each state as it is drawn. Use nice pale pastel colours, otherwise the map looks silly.

8. ***(Advanced Extra Extra Extra Credit)*** Remove Distortions.

      Simple map projections always make regions at higher latitudes look much bigger than they really are. Use your knowledge of geometry and trigonometry to make your program draw the country as it would really appear to someone in a rocket thousands of miles above the Earth's surface.

Of course the data file doesn't contain the information you'd need to show the shading as illustrated here, not does it give the outlines for Canada or Mexico, but this does show approximately the kind of non-distorted shape that you should be able to get.

EEN118 LAB NINE

This lab involves some scientific data processing. You will download a Geographical Database File which contains the coordinates of the boundaries of the 48 connected states and the five great lakes, and write a program that draws maps requested by the user.

Download the file "`usa.map`" from the class web site. The format of the file is very simple; it describes the outlines of the 48 older states plus the five great lakes. Here is the beginning of the file:

```
WA
122750   49000
117000   49000
117000   46333
116866   46000
119000   46000
122666   45500
122916   46166
123916   46250
124166   47000
124666   48333
122833   48166
122666   47416
122583   47333
122416   47333
122750   49000
-1 -1
ID
117000   49000
116000   49000
116000   48000
114500   46500
. . .
```

and it continues like that for a long time. The first line "`WA`" indicates that this is the description of a state: WA is the postal abbreviation for Washington. The next 15 lines give the coordinates of a point along the border of the state. The coordinates are actually longitude and latitude measured in thousandths of a degree, but you can treat them simply as `x` and `y` values. The two `-1`'s after the list of numbers are simply to give you an easy way of telling that the list has finished (no real data in this file is ever negative). Then you see "ID" introducing the next state, Idaho's, description.

The whole file is just like that. First a state's abbreviation, then a list of coordinates, then -1 -1, all repeated 55 times. Notice that the coordinates of the last point are the same as the coordinates of the first point (122750, 49000). This is true of every state's description; they all make nice closed figures. The coordinates 122750, 49000 represent the point 122¾ degrees North of the equator and 49 degrees West of the Greenwich meridian. It is where the U.S.-Canadian border meets the Pacific ocean.

At the end of the file, which looks like this...

```
WV
 82666  38333
 82000  37500
 81500  37250
 80250  37500
 79666  38666
 79333  38500
 78333  39500
 77833  39166
 77750  39333
 78166  39583
 78833  39500
 79500  39333
 79500  39666
 80500  39666
 80500  40666
 80666  40666
 81000  39500
 82666  38333
 -1 -1
 END
```

... you'll find the word END. The last state, West Virginia's description ends as usual with -1 -1, then the word END appears. No state has "END" as its postal abbreviation. There is no data following the "END".

For your information, these are the extreme data values that appear in the file:

Minimum Longitude (x)	70750
Maximum Longitude (x)	124666
Minimum Latitude (y)	25166
Maximum Latitude (y)	49333

The states all have their standard two-letter postal abbreviations. The great lakes are given three-letter abbreviations: LKE, LKH, LKM, LKO, and LKS.

1. Draw Washington

Write a program that opens a reasonably large graphics window (you decide the size), and draws the outline of the first state that appears in the data file (i.e. Washington). You will have to scale and shift the coordinates before plotting, as one of the points in Washington is (117000, 49000) and there's no way you're going to get a window that big. Make sure your picture comes out the right way round:

2. Draw any State

    Modify your program so that it asks the user to enter a state's (or lake's) abbreviation, and then draws that state. It should not draw anything else, just the outline of the selected state. Do not worry about sizing the window to fit that state properly. Make the window big enough to draw the whole country, and just draw the one state in its correct position.

3. Did you get it Right?

    Two of the states, Michigan (MI) and Virginia (VA) are not contiguous; they come in two sections separated from each other by water. The two sections of these states have their own descriptions in the data file: there are two sections beginning with VA and two beginning with MI. Make sure that if the user requests a non-contiguous state, all of its portions are drawn.

4. Make it Incremental.

    Add a loop to your program. After creating the window big enough to draw the whole country, it should enter the loop, repeatedly asking the user to enter a state abbreviation, and adding the outline of that state to what has already been drawn, so that the user can build up a map of many states if desired. Remember to open the file each time round the loop, and close it before the end of the loop; that way you'll be able to read it afresh from the beginning each time.

5. Finally make it able to do Everything.

    If the user enters ALL as the desired state abbreviation, your program should draw the outlines of all the states (in the same window), so that a map of the whole country appears. Of course the program should still draw individual states if the input is not "ALL".

6. ***(Extra Credit)*** Make it into an Interactive Coordinate Finder.

There is a function in the library called `WaitForMousePress`. It takes two parameters, which must be `int` *variables*. When called, the function waits until the user (you) clicks the mouse somewhere within the graphics window. It then returns, setting the two parameters to the coordinates of the mouse press. This is how you might use the function in a graphics program:

```
int x, y;
WaitForMousePress(x, y);
cout << "Mouse pressed at " << x << "," << y <<"\n";
```

Modify your program so that if the user types `Q` (for Query) instead of a state abbreviation, it allows him or her to click the mouse somewhere on the map, and prints out in a reasonably neat way, the geographical coordinates of the point indicated. That is, you should convert the values of x and y back from pixel coordinates to Longitude and Latitude. If the user clicks somewhere around Miami, the output should be something in the region of:
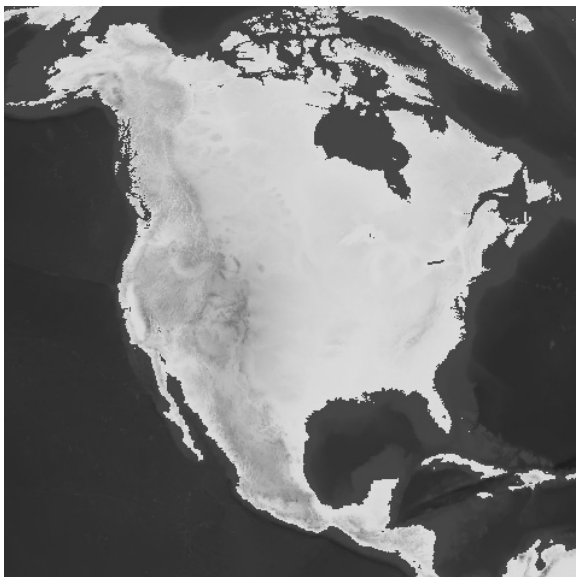
<div align="center">

`80.4 W, 25.8 N`

</div>

7. ***(Extra Credit)*** Colour it in.

If the user enters "MAP", your program should behave exactly as it would if "ALL" had been entered - it should draw the whole country. Except that it should also colour in the interior of each state in a different colour. You don't have to give each state a truly different colour, I don't want you wasting time working out 53 different colours; make it like a real map: use five or six different colours perhaps picking one at random for each state as it is drawn. Use nice pale pastel colours, otherwise the map looks silly.

8. ***(Advanced Extra Extra Extra Credit)*** Remove Distortions.

Simple map projections always make regions at higher latitudes look much bigger than they really are. Use your knowledge of geometry and trigonometry to make your program draw the country as it would really appear to someone in a rocket thousands of miles above the Earth's surface.

Of course the data file doesn't contain the information you'd need to show the shading as illustrated here, not does it give the outlines for Canada or Mexico, but this does show approximately the kind of non-distorted shape that you should be able to get.