Lab 02 - Array Traversal

Instructions:

- **The lab requires completing a few tasks.**

- **Your submissions must be submitted to the Lab02 directory of your GitHub repository or uploaded to the Lab02 assignment on Google classroom.**

- **Accompanying these instructions are a few header files that must be included in the appropriate programs you have to write.**

- **Besides the header files provided, your programs can only include the libraries *iostream*, *string*, *fstream*, *sstream*, and *cctype*.**

- **Cheating of any kind is prohibited and will not be tolerated.**

- **Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the lab.**

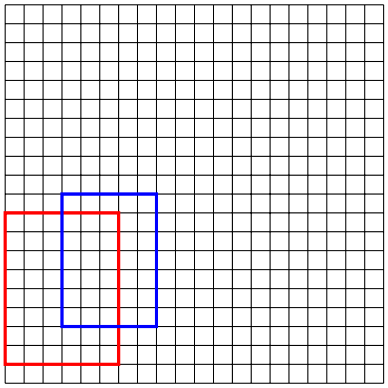
TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, AT THE BEGINNING OF YOUR SUBMISSION(S), ADD A COMMENT THAT CONSISTS OF YOUR NAME AND THE DATE

Grading:

Task	Maximum Points	Points Earned
1	2	
2	2	
3	3	
4	3	
Total	10	

Note: solutions will be provided for tasks colored blue only.

The *Window* class represents a rectangular window that is layered on a 20 by 20 grid. It consists of the xy-coordinate of its lower left-hand corner, its width and its length. Two *Window* objects overlap if they intersect each other. For instance, the *Window* objects A = ((0,1),6,8) and B = ((3,3),5,7) overlap as illustrated in the diagram below



where the red rectangle is object A and the green is object B. The class *Window* contains getter and setter methods for its x and y coordinates, its witdth, its length named `GetX()`, `SetX()`, `GetY()`, `SetY()`, `GetWidth()`, `SetWidth()`, `GetLength()`, and `SetLength()` respectively, and a `ToString()` method that returns a string in the format

((*x*,*y*),*w*,*l*)

where *x*, *y*, *w*, and *l* are the values of its x-coordinate, y-coordinate, width and length respectively. It also overloads the equality operator (==) and ostream operator (<<). Give the above information, use it to define some of the functions for the some of the tasks below.

Task 1

Create the cpp file named `"main.cpp"` that defines a bool function named `HasOverlapped()` whose header is

```
bool HasOverlapped(const Window& a,const Window& b)
```

that returns true if *a* and *b* overlap; otherwise, it returns false.

Task 2

The probability of an event is equal to the count of the event in a set divided by the count of the set. In the same `"main.cpp"` as in task 1, define a double function named `Probability()` whose header is

```
double Probability(const Window& x,const Array<Window>& data)
```

that returns the probability of the elements of *data* that overlaps with *x*.

Task 3

In the same `"main.cpp"` file as the previous tasks, define the *Window* function named `MaxProbability()` whose header is

```
Window MaxProbability(const Array<Window>& data)
```

that returns the element of *data* that has the greatest probability of overlaps with the other elements of *data*. If more than one element share the maximum overlap probability, the element with the earlier index is returned.

Task 4

Create a header file named Project.h that defines the int function named `Preload()` whose header is

```
int Preload(string filename,Array<string>& data)
```

that opens the file whose name is *filename*. It reads the file line by line. If the line is a string of length 36 that consists of only the uppercase characters `'A'` - `'R'` such that there are two of each letter in the string, it adds the string to *data*. It continues this process until either 10 strings are assigned to *data* or it reaches the end of the file. Afterwards, it returns the number of strings that were assigned to *data*. However, if the file does not open, it returns 0.