Problem 1: (Random Walk Robot) A robot is initially located at position $(0, 0)$ in a grid $[-5, 5] \times [-5, 5]$. The robot can move randomly in any of the directions: up, down, left, right. The robot can only move one step at a time.

For each move, print the direction of the move in and the current position of the robot. Use formatted output to print the direction (Down, Up, Left or Right) in the left. The direction takes 10 characters in total and fill in the field with empty spaces. The statement to print results in such format is given below:

```
cout << setw(10) << left << "Down" << ...  ;
cout << setw(10) << left << "Up" << ...;
```

If the robot moves back to the original place (0,0), print "Back to the origin!" to the console and stop the program. If it reaches the boundary of the grid, print "Hit the boundary!" to the console and stop the program. A successful run of your code may look like: Due to

```
Right     (1,0)
Up        (1,1)
Right     (2,1)
Down      (2,0)
Up        (2,1)
Up        (2,2)
Down      (2,1)
Left      (1,1)
Left      (0,1)
Left      (-1,1)
Right     (0,1)
Right     (1,1)
Down      (1,0)
Left      (0,0)
Back to the origin!
```

or

```
Left      (-1,0)
Left      (-2,0)
Right     (-1,0)
Left      (-2,0)
Up        (-2,1)
Up        (-2,2)
Down      (-2,1)
Down      (-2,0)
Left      (-3,0)
Right     (-2,0)
Down      (-2,-1)
Right     (-1,-1)
Down      (-1,-2)
Left      (-2,-2)
Down      (-2,-3)
Down      (-2,-4)
Down      (-2,-5)
Hit the boundary!
```

randomness, your results may have a different trajectory path than examples. Your code should generate different robot trajectories for each run.
Hint: Use two int variables x and y to represent robot's position and use the random integer generator to pick a moving direction at each step. Use a loop to keep moving the robot as long as the robot doesn't hit the boundary or go back to the origin.

Problem 2: (Compute square root without using `sqrt` function) Write a program to locate the square root of an integer. Note: you are not allowed to use `sqrt` or any function in `cmath`.
As $\sqrt{n}$ is the solution to equation $x^2 - n = 0$, evaluating the value of $\sqrt{n}$ can be converted to solving equation $x^2 - n = 0$ using bisection. See bisection on `https://en.wikipedia.org/wiki/Bisection_method`.
The program promotes users to enter an integer n and prints intervals that $\sqrt{n}$ is located in after cutting one interval in halves.

- Use $[0, n]$ as the initial interval.
- The stopping interval has a length less than `TOL = 1e-5`.

Hints: use two doubles `left` and `right` to record two end points of the current interval. Use `middle` to record the middle point of the current interval, then the square root will locate in either `[left, middle]` or `[middle, right]`. Which interval to choose is determined by the Boolean value of `(middle*middle-n>0)`. Write a loop to repeat the previous steps until the interval's length is less than `TOL = 1e-5`.

Here are the samples when user enters 5 and 4. You can see the final interval length is less than the tolerance `TOL = 1e-5`.

```
Please enter a positive integer: 5
Square root of 5 is located in the interval [0, 5].
Square root of 5 is located in the interval [0, 2.5].
Square root of 5 is located in the interval [1.25, 2.5].
Square root of 5 is located in the interval [1.875, 2.5].
Square root of 5 is located in the interval [2.1875, 2.5].
Square root of 5 is located in the interval [2.1875, 2.34375].
Square root of 5 is located in the interval [2.1875, 2.265625].
Square root of 5 is located in the interval [2.2265625, 2.265625].
Square root of 5 is located in the interval [2.2265625, 2.24609375].
Square root of 5 is located in the interval [2.2265625, 2.236328125].
Square root of 5 is located in the interval [2.231445312, 2.236328125].
Square root of 5 is located in the interval [2.233886719, 2.236328125].
Square root of 5 is located in the interval [2.235107422, 2.236328125].
Square root of 5 is located in the interval [2.235717773, 2.236328125].
Square root of 5 is located in the interval [2.236022949, 2.236328125].
Square root of 5 is located in the interval [2.236022949, 2.236175537].
Square root of 5 is located in the interval [2.236022949, 2.236099243].
Square root of 5 is located in the interval [2.236061096, 2.236099243].
Square root of 5 is located in the interval [2.236061096, 2.23608017].
Square root of 5 is located in the interval [2.236061096, 2.236070633].
The interval length is 9.536743164e-06.
```

```
Please enter a positive integer: 4
Square root of 4 is located in the interval [0, 4].
Square root of 4 is located in the interval [2, 4].
Square root of 4 is located in the interval [2, 3].
Square root of 4 is located in the interval [2, 2.5].
Square root of 4 is located in the interval [2, 2.25].
Square root of 4 is located in the interval [2, 2.125].
Square root of 4 is located in the interval [2, 2.0625].
Square root of 4 is located in the interval [2, 2.03125].
Square root of 4 is located in the interval [2, 2.015625].
Square root of 4 is located in the interval [2, 2.0078125].
Square root of 4 is located in the interval [2, 2.00390625].
Square root of 4 is located in the interval [2, 2.001953125].
Square root of 4 is located in the interval [2, 2.000976562].
Square root of 4 is located in the interval [2, 2.000488281].
Square root of 4 is located in the interval [2, 2.000244141].
Square root of 4 is located in the interval [2, 2.00012207].
Square root of 4 is located in the interval [2, 2.000061035].
Square root of 4 is located in the interval [2, 2.000030518].
Square root of 4 is located in the interval [2, 2.000015259].
Square root of 4 is located in the interval [2, 2.000007629].
The interval length is 7.629394531e-06.
```

Instructions:

- (5pt) Name your files `randomWalk.cpp` and `mySqrt.cpp` and submit them to CCLE. Add description of this file in the beginning to show your ownership.

- (10pt) Write your code with good coding practices, including commenting your code, using descriptive variable names, using constant variables, etc.
- (Problem 1 45pt & Problem 2 40pt) Code compiles with Visual Studio 2019 and solves the questions. Students may lose the majority of points if their code doesn't compile with VS 2019. Students should test their work with Apporto virtual machines if they don't have Visual Studio 2019 available on their own computer. Please manually log out your account after using the virtual machine.

  To receive full credits, the output must look EXACTLY the same as instructed above, including words, spaces, symbols, etc. Your code should not only work for the above examples, but also work for other different inputs.