Assignment Number: 2

Problem statement:

Random walk: A (drunken) cockroach is placed on a given square in the middle of a tile floor in a rectangular room of size n x m tiles. The bug wanders (possibly in search of an aspirin) randomly from tile to tile throughout the room. Assuming that it may move from his present tile to any of the eight tiles surrounding it (unless it is against a wall) with equal probability, how long will it take him to touch every tile on the floor at least once?
Write a C++ program to graphically show a random walk of a (drunken) cockroach and find the no of moves made.

Objectives:

- Understand the use of array.
- Understand random walk problems.

Theory:

Random Walk Problems:
Random walk concept is often used to simulate the movement of the particles or creatures in physics and computer science.

The Drunken Cockroach Scenario:
Imagine a drunken cockroach placed on a square within a rectangular room. The room is defined by its dimensions, represented as an n×m grid of tiles. The intoxicated bug wanders randomly, potentially in search of an aspirin, moving from one tile to another throughout the room.
The cockroach has the freedom to move from its current tile to any of the eight adjacent tiles (unless it is against a wall). This means it can move up, down, left, right, and diagonally with equal probability, exploring the room until it has touched every tile at least once.

That is, if (i, j) is the initial position of the cockroach then the cockroach can only move on the tiles as shown in the following figure-

i--,j++	i,j++	i++,j++
i--,j	i,j	i++,j
i--,j--	i,j--	i++,j--

Figure 1

So there are 8 possible cases for the cockroach to move.
For that we will keep on generating a random number from 1 to 8 and based on the number generated we can implement these above cases (using switch case) and make the cockroach move the tile.

Generating random number in C++
Unlike python it is little trickier to generate random number in C++.
We have to use srand(time(0)) in the main function in order to generate random number every time. Both rand() and srand() functions are present in cstdlib header whereas time() in ctime header file.

Lets see a program to generate the random number in C++

```
//include <cstdlib> and <ctime>
//generate between lb-ub

srand(time(0))
int lb=20,ub=45;                        //including 20 and 45
for (int i=0;i<25;i++)
{
   cout<<(rand()%(ub-lb+1)+lb)<<endl;
}
```

This will print 25 numbers in between 20 and 45 (including 20 and 45).

- ➔ To check whether move is valid or not just put condition

```
If(0<i<n && 0<j<m):
        Valid points
Else:
        Generate new points(i,j)
```

- ➔ To check whether matrix is filled or not there are many ways. Choose suitable logic according to your preference.
  For example one of the logic could be:

  Make all elements of the matrix 0.
  Keep on incrementing matrix[i][j] if the point is valid.
  And at last traverse the matrix. If 0 found then matrix is not full.

  Above logic is easy to understand but increases the time complexity of our program.

Algorithm:

Step 1: Start

Step 2: Read dimensions of the matrix(Tiles)

Step 3: Read initial position of the cockroach

Step 4: initialise all the elements in matrix =0

Step 5: while(matrix_is_filled)

- Generate random number and check whether it is valid move or not.
- If valid :
  Totalmoves++
  Else :
  Again generate number

Step 6: Display total moves

Step 7: Stop

Time complexity: Discuss the time complexity of following function with respect to best case and worst case

Practice problem:
Write possible cases for a knight to move on the chess board.(See figure 1)