1) RAT IN THE MAZE PROBLEM:

METHOD :
This sum is solved using the recursion.We send the recursion in the 4 direction depending on whether that box is earlier visited in the same path of not.
For marking the Box as visited we make it equal to 2.And then after coming out from the recursion we again make it equal to 1.
Link Of Explanation : Search in a Maze| Rat in a Maze | Graph | Love Babbar DSA Sheet [Expla…

CODE FOR THE ABOVE METHOD:

```
class Solution{
  public:
  void function(vector<vector<int>> &m,int i,int j,int n,string s,vector<string> &answer){
    if(i<0 || j<0 || i>=n || j>=n){
      return;
    }
    else if(i==n-1 && j==n-1){
      answer.push_back(s);
    }
    else if(m[i][j]==0){
      return;
    }
    else{
      m[i][j]=2;
      if(i-1>=0 && m[i-1][j]!=2){
        function(m,i-1,j,n,s+'U',answer);
      }
      if(i+1<n && m[i+1][j]!=2){
        function(m,i+1,j,n,s+'D',answer);
      }
      if(j-1>=0 && m[i][j-1]!=2){
        function(m,i,j-1,n,s+'L',answer);
      }
      if(j+1<n && m[i][j+1]!=2){
        function(m,i,j+1,n,s+'R',answer);
      }
      m[i][j]=1;
    }
  }
  vector<string> findPath(vector<vector<int>> &m, int n) {
    vector<string> answer(0);
    if(m[n-1][n-1]==0){
      return answer;
    }
    else{
      function(m,0,0,n,"",answer);
      return answer;
    }
```

```
    }
};
```