17) PRINT ALL THE POSSIBLE PATHS FROM TOP LEFT TO BOTTOM RIGHT

METHOD:

1) This sum is also the same as that of the Rat in the maze problem.
2) Here we pass the recursion only in 2 direction that is in the right direction and the bottom direction.
3) Here the recursion will pass compulsorily in both the direction as there is no condition for the block of the position.

CODE OF THE PROGRAM:

```
#include<iostream>
#include<vector>
#include<limits.h>
using namespace std;

void display(vector<int> &t){
   cout<<"\n ";
   for(int i=0;i<t.size();i++){
      cout<<t[i];
   }
}

void fun(vector<vector<int>> & grid,int r,int c,int currr,int currc,vector<int> &t){
   if(currr==r-1 && currc==c-1){
      t.push_back(grid[currr][currc]);
      display(t);
      t.pop_back();
      return;
   }
   else if(currr<0 || currr>=r || currc<0 || currc>=c){
      return;
   }
   else{
      t.push_back(grid[currr][currc]);
      fun(grid,r,c,currr,currc+1,t);
      fun(grid,r,c,currr+1,currc,t);
      t.pop_back();
   }
}

int main(){
   int r,c;
   cout<<"\n Enter the number of rows in the grid:";
   cin>>r;
   cout<<"\n Enter the number of columns in the grid:";
   cin>>c;
   vector<vector<int>> grid(r,vector<int>(c));
   cout<<"\n Enter the elements in the grid:\n";
```

```
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            cin>>grid[i][j];
        }
    }
    vector<int> t;
    cout<<"\n The answers : ";
    fun(grid,r,c,0,0,t);
    return 0;
}
```