# Number of paths

The problem is to count all the possible paths from top left to bottom right of a MxN matrix with the constraints that from each cell you can either move to right or down.

**Example :**

**Input:**
M = 3 and N = 3
**Output:** 6
**Explanation:**
Let the given input 3*3 matrix is filled
as such:
A B C
D E F
G H I
The possible paths which exists to reach
'I' from 'A' following above conditions
are as follows:ABCFI, ABEHI, ADGHI, ADEFI,
ADEHI, ABEFI
**Expected Time Complexity:** O(m + n - 1))

**Expected Auxiliary Space:** O(1)

**Constraints:**

1 ≤ M, N ≤ 10

```cpp
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int no_of_path(int n,int m)
{
  if(n==1||m==1)
     return 1;
  return (no_of_path(n-1,m)+no_of_path(n,m-1));
}
void solve()
{
  int n,m;
  cin>>n>>m;
  cout<<no_of_path(n,m);
}
int main()
{
      int t;
  cin>>t;
  while(t--)
  {
      solve();
     cout<<endl;
  }
```