**Number of paths**

[recursion](http://www.practice.geeksforgeeks.org/tag-page.php?tag=recursion&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)

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 only to right or down.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is M and N, M is number of rows and N is number of columns.  
  
**Output:**

Print the number of paths.  
  
**Constraints:**

1 ≤ T ≤ 30  
1 ≤ M,N ≤ 10  
  
**Example:**

Input  
2  
3 3  
2 8

Output  
6  
8

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=474>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

using namespace std;

int main() {

    int T;

    scanf("%d", &T);

    while(T--) {

        int M, N;

        scanf("%d %d", &M, &N);

        int paths[M][N];

        for(int i =0; i < N; i++) {

            paths[0][i] = 1;

        }

        for(int i =0; i<M; i++){

           paths[i][0] = 1;

        }

        for(int i = 1; i<M; i++) {

            for(int j = 1; j<N; j++) {

                paths[i][j] = paths[i-1][j] + paths[i][j-1];

            }

        }

*/\**

*for(int i = 0; i<M; i++) {*

*for(int j = 0; j<N; j++) {*

*printf("%d ", paths[i][j]);*

*}*

*printf("\n");*

*}\*/*

         printf("%d**\n**", paths[M-1][N-1]);

    }

  system("pause");

 return 0;

}