**Number of Ways**Problem Code: **NWAYS**

******

[Submit (Practice)](https://www.codechef.com/submit/NWAYS)

**Read problems statements in**[**Mandarin Chinese**](http://www.codechef.com/download/translated/LTIME24/mandarin/NWAYS.pdf)**and**[**Russian**](http://www.codechef.com/download/translated/LTIME24/russian/NWAYS.pdf)**.**

Devu is learning Combinatorics in his college. He find it very interesting to calculate number of ways of going to point **(c,d)** from point **(a,b)** in co-ordinate plane. We can take horizontal and vertical steps only and can not visit at a point twice. In a step, you can move one unit only. We have to reach to the point **(c,d)** from the point **(a,b)** using **abs(a-c)+ abs(b-d)** steps only.

Devu has two sets of points. Set **A** contains points having **X** co-ordinate **0** and **Y** co-ordinates varying from **1** to **N**(both inclusive). Set **B** contains points having **X** co-ordinate **K** and **Y** co-ordinates varying from **1** to **N**(both inclusive). Both sets contains **N** number of integral points. He wants to calculate the sum of number of ways to going to the each point of set **B** from the each point of set **A** .

As the answer can be large, print it modulo **1000000007**.

**Input**

* First line of input contains an integer **T** denoting number of test cases.
* Next **T** lines will contain pair of integers **N** and **K**

**Output**

For each test case, print a single integer representing the answer of that test case.

**Constraints**

**Subtask #1: 10 points**

* + **1 ≤ T ≤ 20, 1 ≤ N ,K ≤ 1000**

**Subtask #1: 10 points**

* + **1 ≤ T ≤ 20, 1 ≤ N ,K ≤ 106**

**Subtask #3: 80 points**

* + **1 ≤ T ≤ 10000, 1 ≤ N,K ≤ 106**

**Example**

**Input:**

2

2 2

4 5

**Output:**

8

236

**Explanation**

For the first sample case,  
  
ways[(0,1)->(2,1)]= 1  
  
ways[(0,2)->(2,2)]= 1  
  
ways[(0,1)->(2,2)]= 3  
  
ways[(0,2)->(2,1)]= 3

Find the number of ways to move from all points in set X = \{(0,i): 1 \leq i \leq n \}*X*={(0,*i*):1≤*i*≤*n*} to all point in set Y = \{(k,i): 1 \leq i \leq n \}*Y*={(*k*,*i*):1≤*i*≤*n*} using minimum number of steps.

**EXPLANATION:**

We are given two sets of points **X** and **Y** (described above) in 2-D grid, and we have to find the number of ways to move from all points in X to all points in Y using minimum number of steps. Note that only vertical and horizontal travelling is allowed and since we have to move from one point to another in minimum number of steps, we cannot visit a point twice.

[The number of shortest paths 27](http://math.stackexchange.com/questions/103470/how-can-i-find-the-number-of-the-shortest-paths-between-two-points-on-a-2d-latti) from *(0,0)* to *(m,n)* in 2D grid is m+n \choose n(*nm*+*n*​). Using this result, in this problem, we essentially have to find the following term in which the first summation loops over the points in set **X** and second summation loops over the points in set **Y**.

S = \sum\_{i=1}^n \sum\_{j=1}^n {|j-i|+k \choose k}*S*=*i*=1∑*n*​*j*=1∑*n*​(*k*∣*j*−*i*∣+*k*​)

Now the absolute value of j-i*j*−*i* is a bug and we have to some how get rid of that absolute sign. The easiest way to do so is to consider the case when j \geq i*j*≥*i* and j < i*j*<*i* . So, for any point *i* in set **X**, divide the points in set **Y** into 3 parts - lower y-points , upper y-points and y-point which is at the same level. Note that the upper and lower point cases are symmetrical. Hence, S*S* can be written as follows

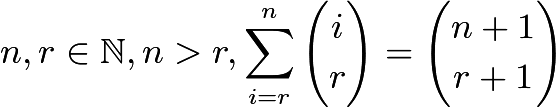
S = 2\sum\_{i=1}^n \sum\_{j=i}^n {j-i+k \choose k} - n*S*=2*i*=1∑*n*​*j*=*i*∑*n*​(*kj*−*i*+*k*​)−*n*

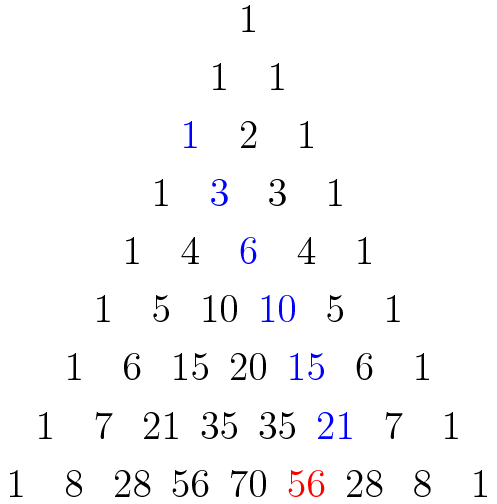
Now the only thing which is left, is to calculate \sum\_{i=1}^n \sum\_{j=i}^n {j-i+k \choose k}∑*i*=1*n*​∑*j*=*in*​(*kj*−*i*+*k*​).

\begin{aligned} \sum\_{i=1}^n \sum\_{j=i}^n {j-i+k \choose k} &= \sum\_{i=1}^n \sum\_{j=k}^{n-i+k} \binom{j}{k} \\ &= \sum\_{i=1}^n \binom{n-i+k+1}{k+1} \qquad \text{ Hockey Stick Identity } \\ &= \sum\_{i=k+1}^{n+k} \binom{i}{k+1} \\ &= \binom{n+k+1}{k+2} \qquad \qquad \quad \ \text{ Hockey Stick Identity } \end{aligned}*i*=1∑*n*​*j*=*i*∑*n*​(*kj*−*i*+*k*​)​=*i*=1∑*n*​*j*=*k*∑*n*−*i*+*k*​(*kj*​)=*i*=1∑*n*​(*k*+1*n*−*i*+*k*+1​) Hockey Stick Identity =*i*=*k*+1∑*n*+*k*​(*k*+1*i*​)=(*k*+2*n*+*k*+1​)  Hockey Stick Identity ​

The binomial terms can be calculated in \mathcal{O}(1)O(1) time if we have precomputed the factorial and inverse factorial. Hence, answer for every test case can be computed in \mathcal{O}(1)O(1) time.

## Hockey-Stick Identity

For .



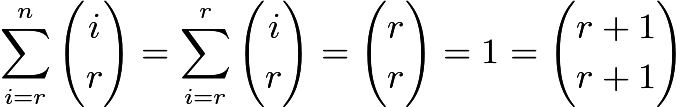
This identity is known as the *hockey-stick* identity because, on Pascal's triangle, when the addends represented in the summation and the sum itself is highlighted, a hockey-stick shape is revealed.

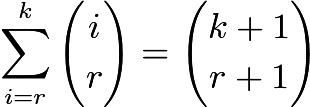
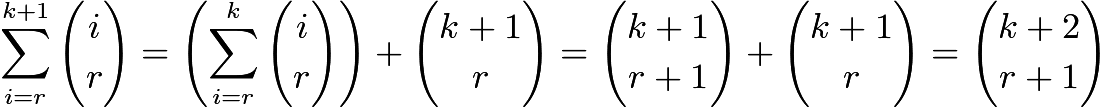
### Proof

**Inductive Proof**

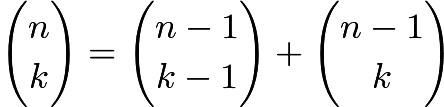
This identity can be proven by induction on $n$.

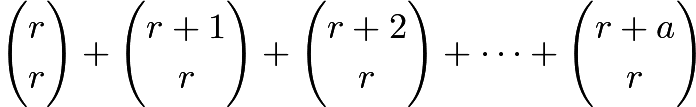
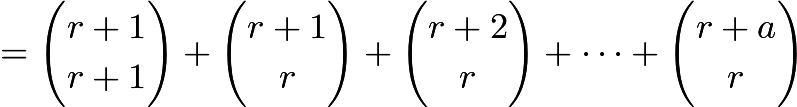
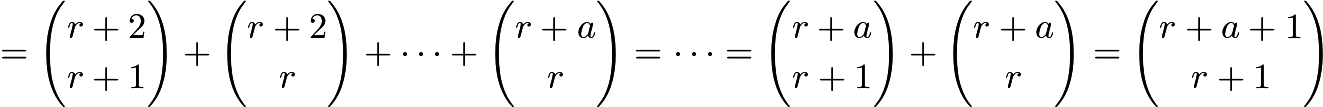
Base Case Let $n=r$.

.

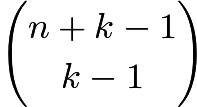
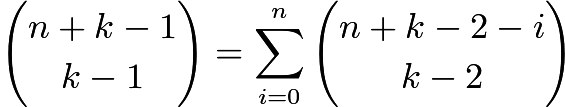
Inductive Step Suppose, for some $k\in\mathbb{N}, k>r$, . Then .

**Algebraic Proof**

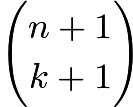
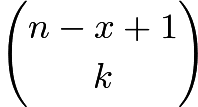
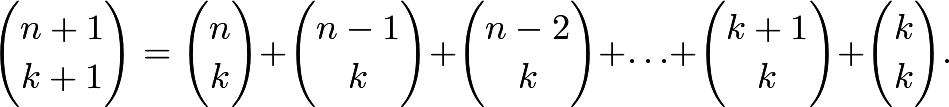
It can also be proven algebraically with [Pascal's Identity](https://artofproblemsolving.com/wiki/index.php/Pascal%27s_Identity), . Note that

  , which is equivalent to the desired result.

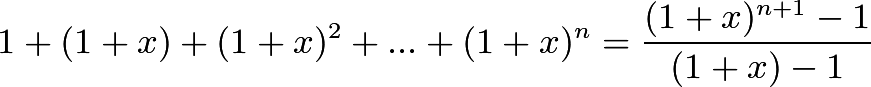
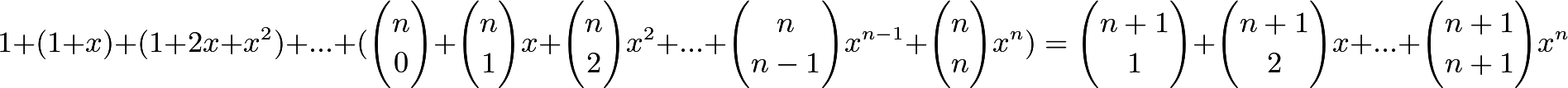
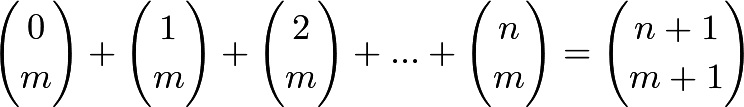
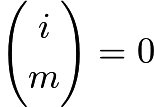
**Combinatorial Proof 1**

Imagine that we are distributing $n$ indistinguishable candies to $k$ distinguishable children. By a direct application of Balls and Urns, there are  ways to do this. Alternatively, we can first give $0\le i\le n$ candies to the oldest child so that we are essentially giving $n-i$ candies to $k-1$ kids and again, with Balls and Urns, , which simplifies to the desired result.

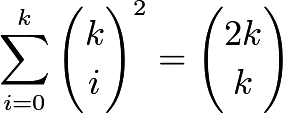
**Combinatorial Proof 2**

We can form a committee of size $k+1$ from a group of $n+1$ people in  ways. Now we hand out the numbers $1,2,3,\dots,n-k+1$ to $n-k+1$ of the $n+1$ people. We can divide this into $n-k+1$ disjoint cases. In general, in case $x$, $1\le x\le n-k+1$, person $x$ is on the committee and persons $1,2,3,\dots, x-1$ are not on the committee. This can be done in  ways. Now we can sum the values of these $n-k+1$ disjoint cases, getting

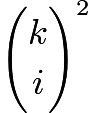
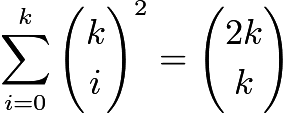
**Algebraic Proof 2**

Apply the finite geometric series formula to $(1+x)$:Then expand with the Binomial Theorem and simplify:Finally, equate coefficients of $x^m$ on both sides:Since for $i<m$, , this simplifies to the hockey stick identity. -- EVIN-

## Another Identity



### Hat Proof

We have $2k$ different hats. We split them into two groups, each with k hats: then we choose $i$ hats from the first group and $k-i$ hats from the second group. This may be done in  ways. Evidently, to generate all possible choices of $k$ hats from the $2k$ hats, we must choose $i=0,1,\cdots,k$ hats from the first $k$ and the remaining $k-i$ hats from the second $k$; the sum over all such $i$ is the number of ways of choosing $k$ hats from $2k$. Therefore , as desired.

CODE:

#include<bits/stdc++.h>

#define pb push\_back

#define int long long int

#define vec vector<int>

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int mod=1e9+7;

int fact[3000001];

void factorial()

{

fact[1]=1,fact[0]=1;

for(int i=2;i<3000001;i++)

fact[i]=((i%mod)\*(fact[i-1]%mod))%mod;

}

int inverse(int n)

{

n%=mod;

int res=1,b=mod-2;

while(b>0)

{

if(b&1)

res=((res%mod)\*(n%mod))%mod;

n=((n%mod)\*(n%mod))%mod;

b>>=1;

}

return res%mod;

}

main()

{

factorial();

int t;

cin>>t;

while(t--)

{

int n,k;

cin>>n>>k;

cout<<((2\*((fact[n+k+1]%mod)\*((inverse(fact[k+2])%mod)\*(inverse(fact[n-1])%mod)%mod)))%mod +mod -n)%mod<<"\n";

}

}