There are N buildings standing in a straight line in the City, numbered from 1 to N. The heights of all the buildings are distinct and between 1 and N. You can see F buildings when you standing in front of the first building and looking forward, and B buildings when you are behind the last building and looking backward. A building can be seen if the building is higher than any building between you and it.   
Now, given N, F, B, your task is to figure out how many ways all the buildings can be.

Input

First line of the input is a single integer T (T<=100000), indicating there are T test cases followed.   
Next T lines, each line consists of three integer N, F, B, (0<N, F, B<=2000) described above.

Output

For each case, you should output the number of ways mod 1000000007(1e9+7).

Sample Input

2

3 2 2

3 2 1

Sample Output

2

1

题目大意：

给你一个n，表示有n个高度分别为1，2，3……n的楼，然后要求你排列这n个楼的位置，使得从最左端看能看到x个楼，从最右端看到y个楼，问你满足要求的方案数。

数组组数1<=t<=100000,1<=n,x,y<=2000

最高的房子为中间，左边有f-1个房子可以看到，右边有b-1个房子，也就是总共选出f+b-2个房子，剩余的房子在它的左边或右边，可以理解为分成了f+b-2组，且含有这个指定的顺序，看成第一类斯特林数，再从f+b-2组里面选出f-1组，答案就是：c[f+b-2][f-1]\*s(n-1,f+b-2)

#include <iostream>

#include <cstdio>

#include <cstring>

#include <cmath>

#include <algorithm>

using namespace std;

typedef long long ll;

const ll mod=1000000007;

const int maxn=2100;

ll c[maxn][maxn],s[maxn][maxn];

void ini()

{

for(int i=0;i<maxn;i++)

{

c[i][0]=c[i][i]=1;

for(int j=1;j<i;j++)

c[i][j]=(c[i-1][j-1]+c[i-1][j])%mod;//杨辉三角dp求组合数

}

for(int i=0;i<maxn;i++)

{

s[i][0]=0;

s[i][i]=1;

for(int j=1;j<i;j++)

s[i][j]=(s[i-1][j-1]+(i-1)\*s[i-1][j])%mod; //第一类

}

}

int n,f,b;

int main()

{

ini();

int T;

scanf("%d",&T);

while(T-- >0)

{

scanf("%d%d%d",&n,&f,&b);

ll ans;

if(f+b-2<maxn)

ans=(c[f+b-2][f-1]\*s[n-1][f+b-2])%mod;

else ans=0;

cout<<ans<<endl;

}

return 0;

}