/\*

动态规划：

令dp[i][0]表示第i步向下走的决策数，dp[i][1]表示第i步向右走的决策数

则有dp[i][p]=dp[x][p^1] (i-k<=x<i)

易知dp[i][0]=dp[i][1] 所以第二维可以省略

最终结果为∑dp[x]\*2 (N-k<x<=N)

\*/

#include<bits/stdc++.h>

using namespace std;

int T;

int mod=1e9+7;

int main()

{

freopen("journey.in","r",stdin);

freopen("journey.ans","w",stdout);

cin>>T;

while(T--)

{

int n,k;

cin>>n>>k;

int dp[100001]={0};

int sum=1;

dp[1]=1;

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

{

dp[i]=sum;

if(i>k)

{

sum=sum-dp[i-k]+mod;

sum%=mod;

}

sum+=dp[i];

sum%=mod;

}

if(n!=0)cout<<(sum<<1)%mod<<endl;

else cout<<1<<endl;

}

}