

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int c=pow(10,9)+7;

ll binpow(int a,int b,int c ){

    if(b==0)

    return 1;

    ll x=binpow(a,b/2,c);

    if(b%2==1){

        return ((((x%c)\*(x%c))%c)\*(a%c))%c;

    }

    else{

        return ((x%c)\*(x%c))%c;

    }

}

int main(){

int t;

cin>>t;

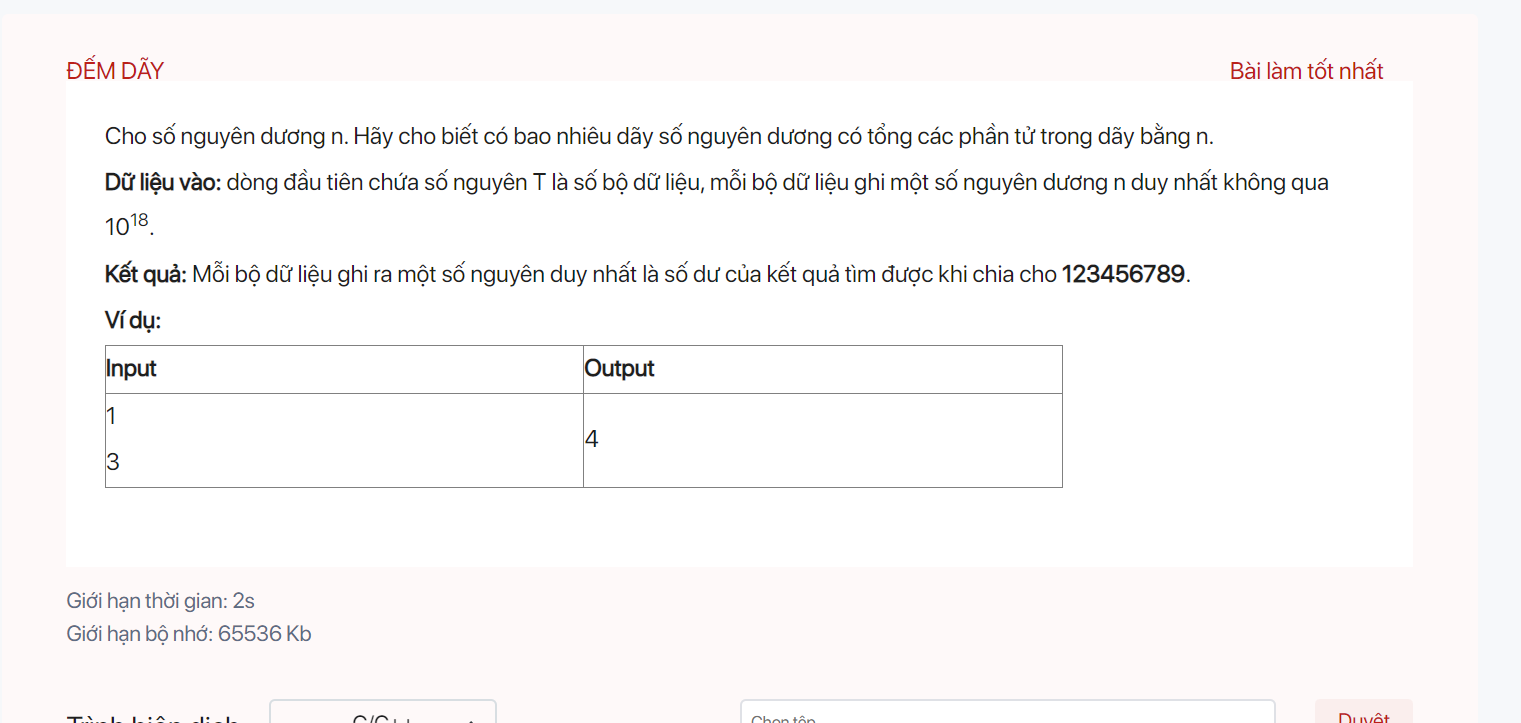
while(t--){

int n,k;

cin>>n>>k;

cout<<binpow(n,k,c)<<endl;

}}



#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll c=123456789;

ll binpow(ll a,ll b ){

    if(b==0)return 1;

    ll x=binpow(a,b/2);

    if(b%2==1){

        return ((((x%c)\*(x%c))%c)\*(a%c))%c;

    }

    else{

        return ((x%c)\*(x%c))%c;

    }

}

int main(){

    int t;

    cin>>t;

    while(t--){

        ll n;

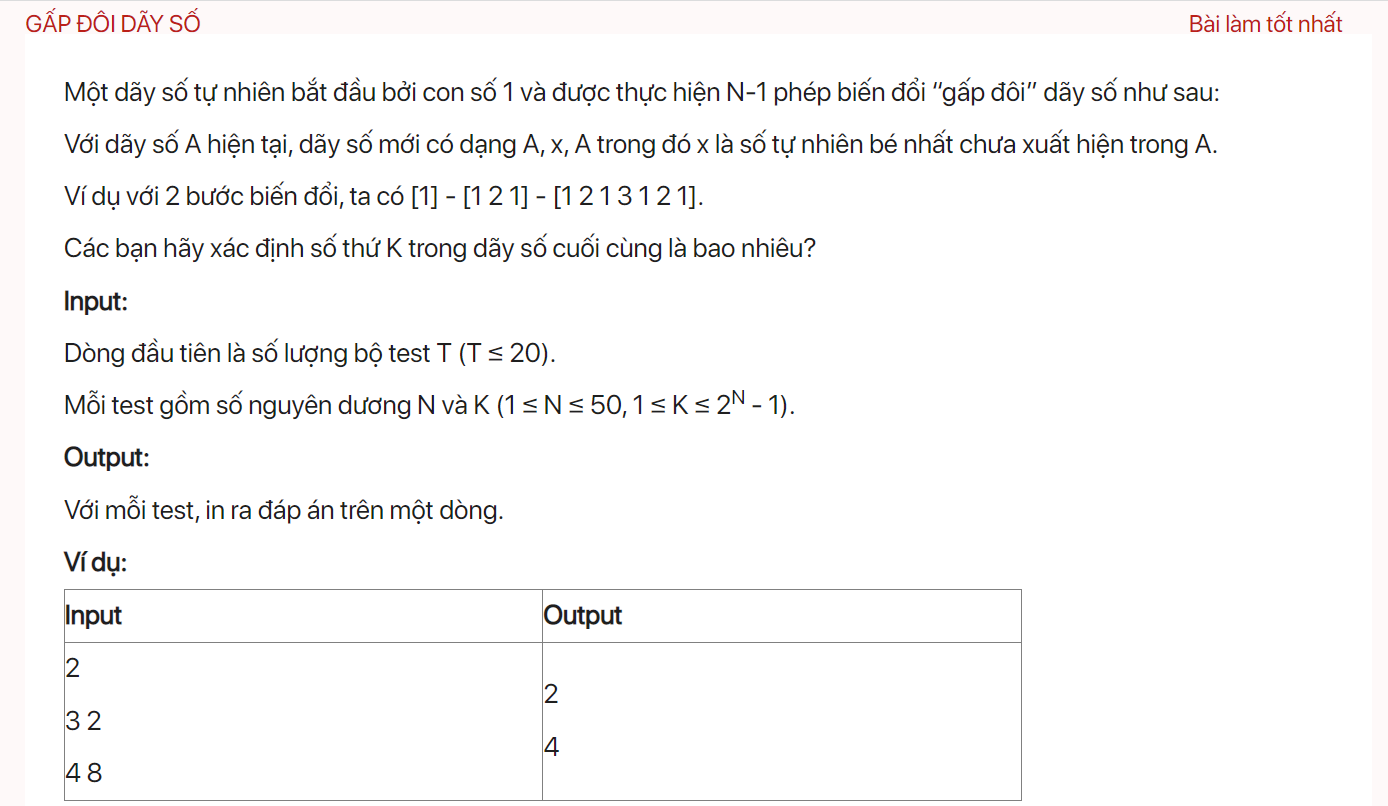
        cin>>n;

        cout<<binpow(2,n-1)<<endl;

    }

}

58.



#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

ll binpow(ll a, ll b)

{

    if (b == 0)

        return 1;

    ll x = binpow(a, b / 2);

    if (b % 2 == 1)

    {

        return x\*x\*a;

    }

    else

    {

        return x\*x;

    }

}

int main()

{

    // bài này so phan tu trung tam cua dãy cuối cùng thu đc là bằng n luôn

    // số phần tử trong dãy cuối cx là 2 mũ n xong trùư 1, khi n=4 thì số phần tử là 15

    // chi so cua phan tu trung tâm  là 2^(n-1) hay suy ra bằng 2 mũ n chia 2

    int t;

    cin >> t;

    while (t--)

    {

        ll n, k;

        cin >> n >> k;

        ll len = binpow(2, n) / 2;

        while (1)

        {

            if (n == 1)

            {

                cout << 1;

                break;

            }

            if (k == len)

            {

                cout << n;

                break;

            }

            if (k > len)

            {

                k = len - (k - len);

            }

            n--;

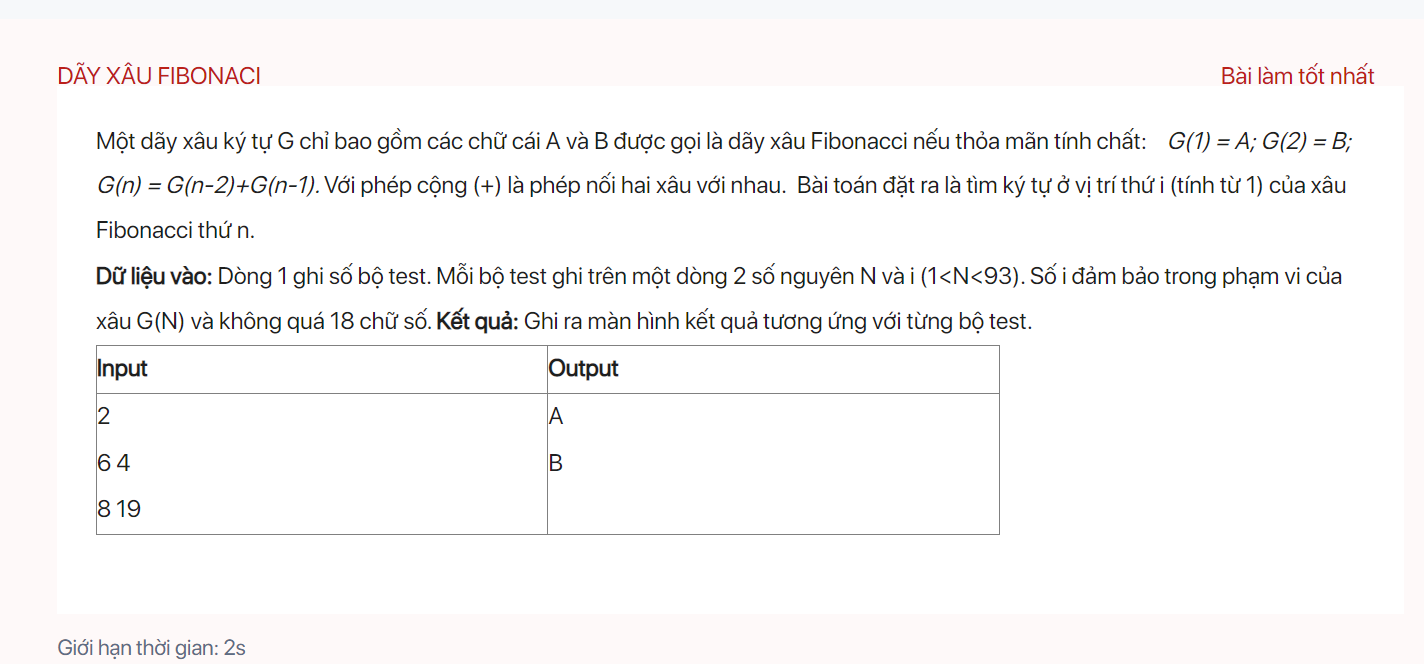
            len = binpow(2, n) / 2;

        }

        cout << endl;

    }

}



#include<bits/stdc++.h>

long long f[93];

using namespace std;

void solve(){

    long long n,i;

    cin>>n>>i;

    while(n>2){

        if(i<=f[n-2]){

            n-=2;

        }

        else{

            i-=f[n-2];

            n-=1;

        }

    }

    if(n==1)cout<<'A';

    else cout<<'B';

    cout<<'\n';

}

int main(){

    f[1]=1;

    f[2]=1;

    for(int i=3;i<=93;i++){

        f[i]=f[i-2]+f[i-1];

    }

    int t;

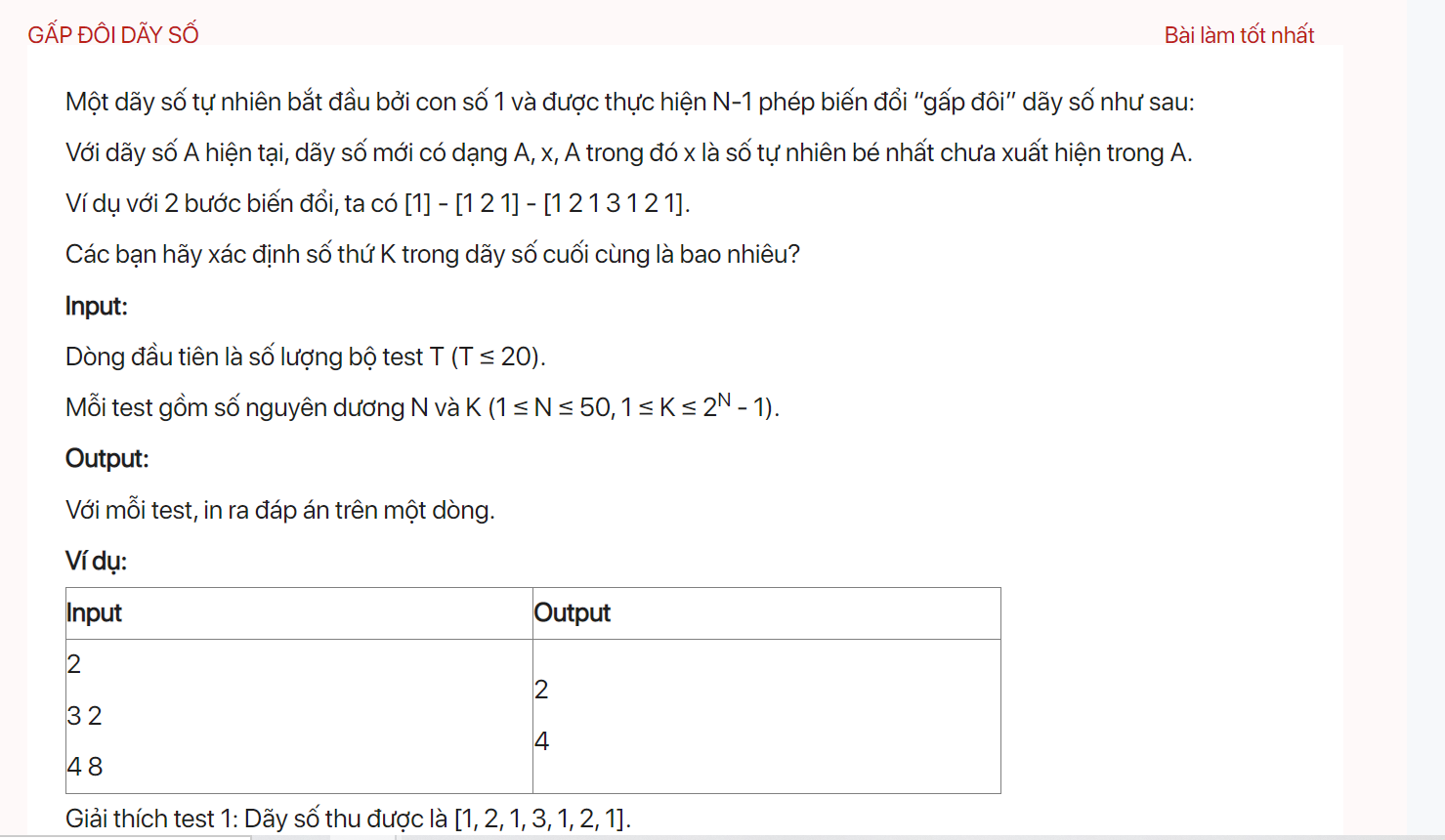
    cin>>t;

    while(t--){

        solve();

    }

}



#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

ll binpow(ll a, ll b)

{

    if (b == 0)

        return 1;

    ll x = binpow(a, b / 2);

    if (b % 2 == 1)

    {

        return x\*x\*a;

    }

    else

    {

        return x\*x;

    }

}

int main()

{

    // bài này so phan tu trung tam cua dãy cuối cùng thu đc là bằng n luôn

    // số phần tử trong dãy cuối cx là 2 mũ n xong trùư 1, khi n=4 thì số phần tử là 15

    // chi so cua phan tu trung tâm  là 2^(n-1) hay suy ra bằng 2 mũ n chia 2

    int t;

    cin >> t;

    while (t--)

    {

        ll n, k;

        cin >> n >> k;

        ll len = binpow(2, n) / 2;

        while (1)

        {

            if (n == 1)

            {

                cout << 1;

                break;

            }

            if (k == len)

            {

                cout << n;

                break;

            }

            if (k > len)

            {

                k = len - (k - len);

            }

            n--;

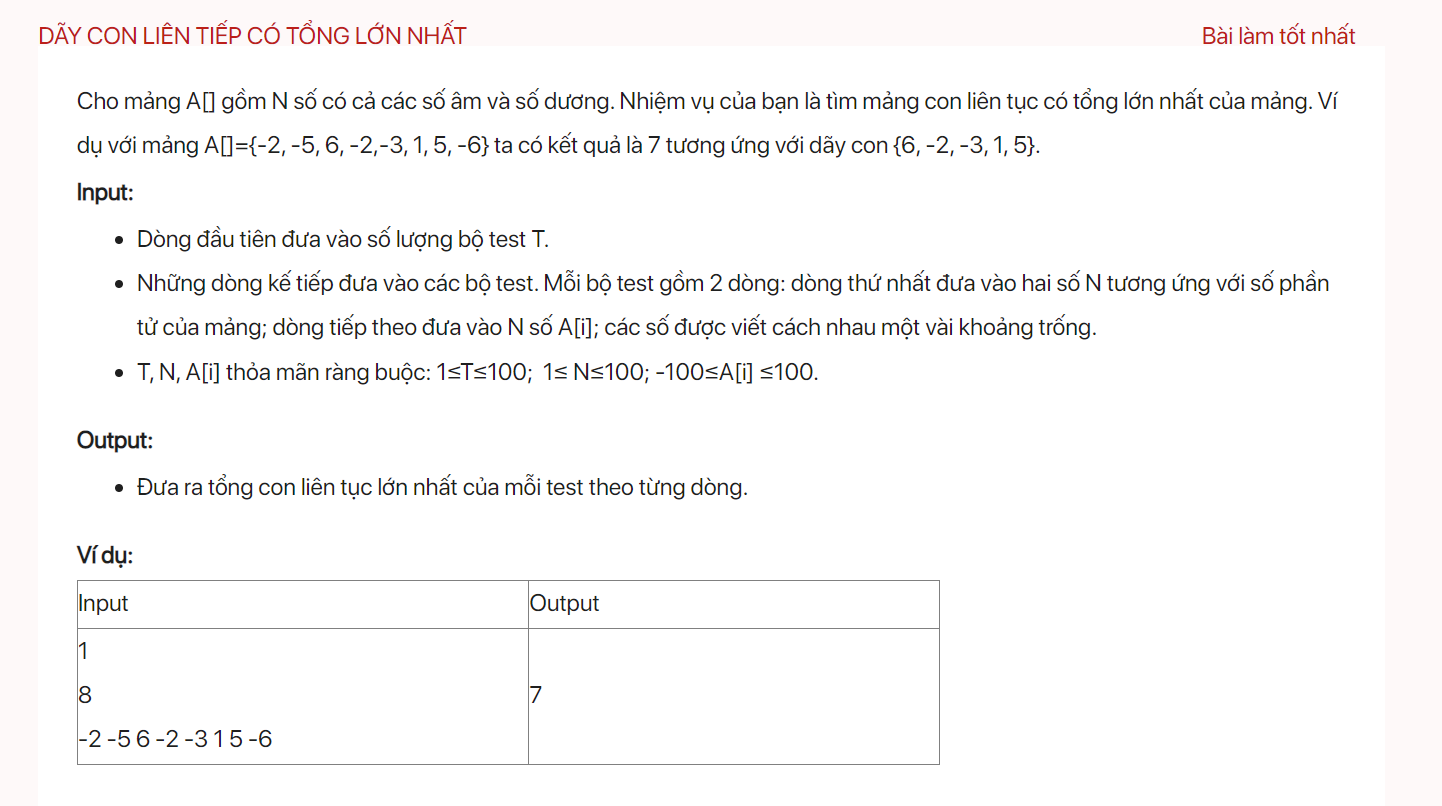
            len = binpow(2, n) / 2;

        }

        cout << endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin>>t;

    while(t--){

        int n;

        cin>>n;

        int a[n];

        for(int i=0;i<n;i++){

            cin>>a[i];

        }

        int s=0;

        int l=-1e9;

        for(int i=0;i<n;i++){

            s+=a[i];

            l=max(s,l);

            if(s<0)

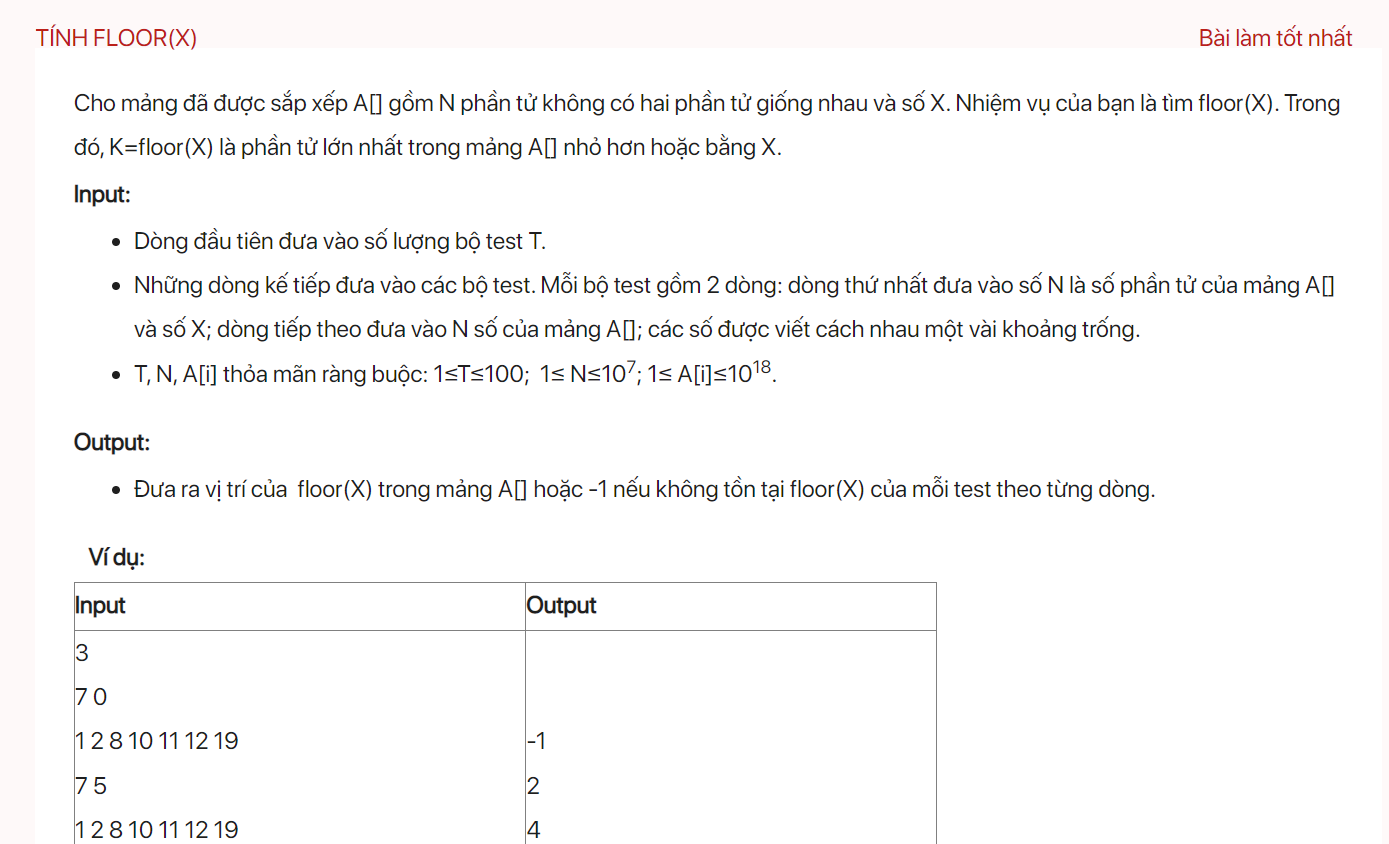
            s=0;

        }

        cout<<l<<endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

int binarysearch(int l,int r,int x,int a[]){

    int res=-1;

    while(l<=r){

        int m=(l+r)/2;

        if(a[m]<=x){

            res=m;

            l=m+1;

        }

        else{

            r=m-1;

        }

    }

    return res;

}

int main(){

    int t;

    cin>>t;

    while(t--){

        int n,x;

        cin>>n>>x;

        int a[n];

        for(int i=0;i<n;i++){

            cin>>a[i];

        }

        if(binarysearch(0,n-1,x,a)==-1)

       cout<<binarysearch(0,n-1,x,a);

       else{

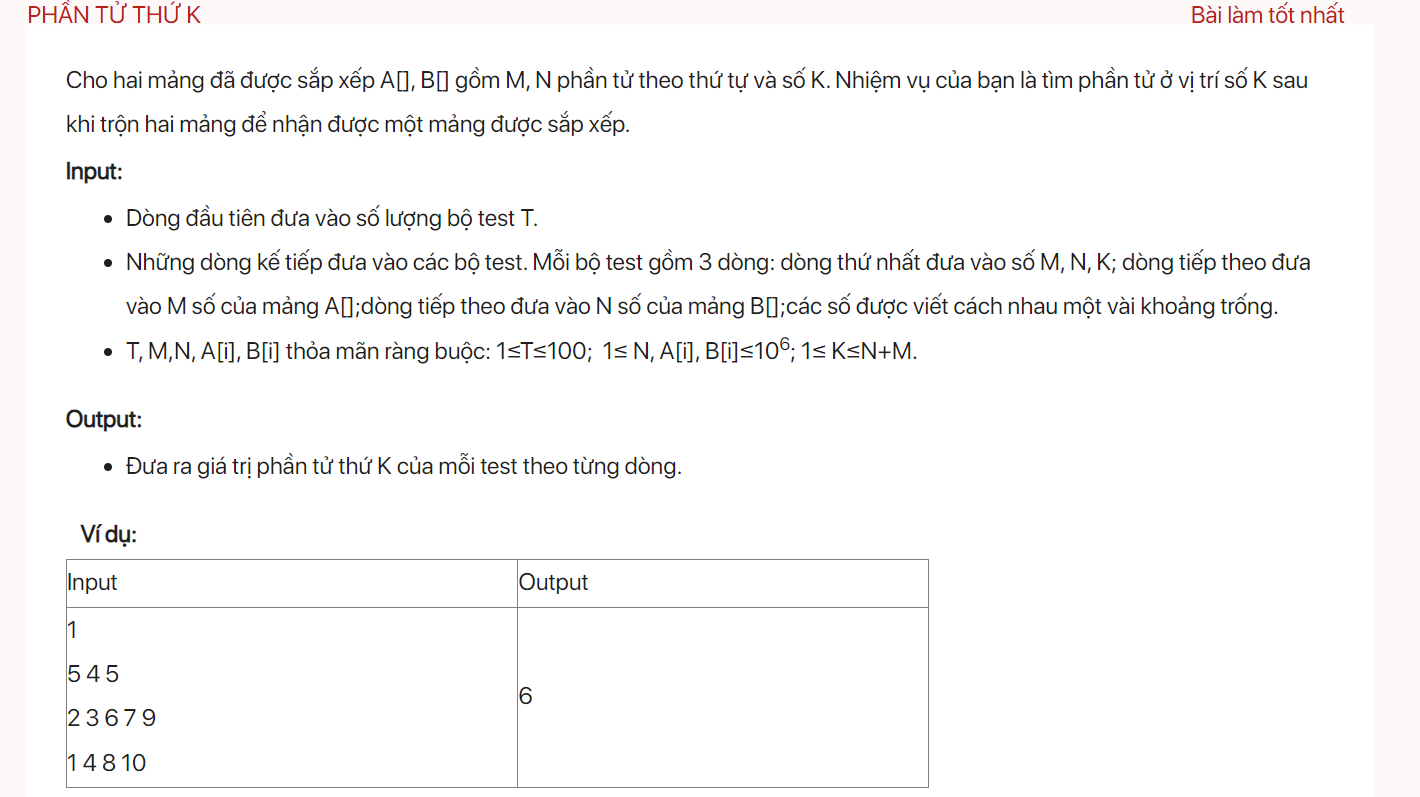
       cout<< binarysearch(0,n-1,x,a)+1;

       }

        cout<<endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin>>t;

    while(t--){

        long long  m,n,k;

        cin>>m>>n>>k;

        int a[n];

        int b[m];

        int c[m+n];

        for(int i=0;i<n;i++){

            cin>>a[i];

            c[i]=a[i];

        }

        for(int i=0;i<m;i++){

            cin>>b[i];

            c[n+i]=b[i];

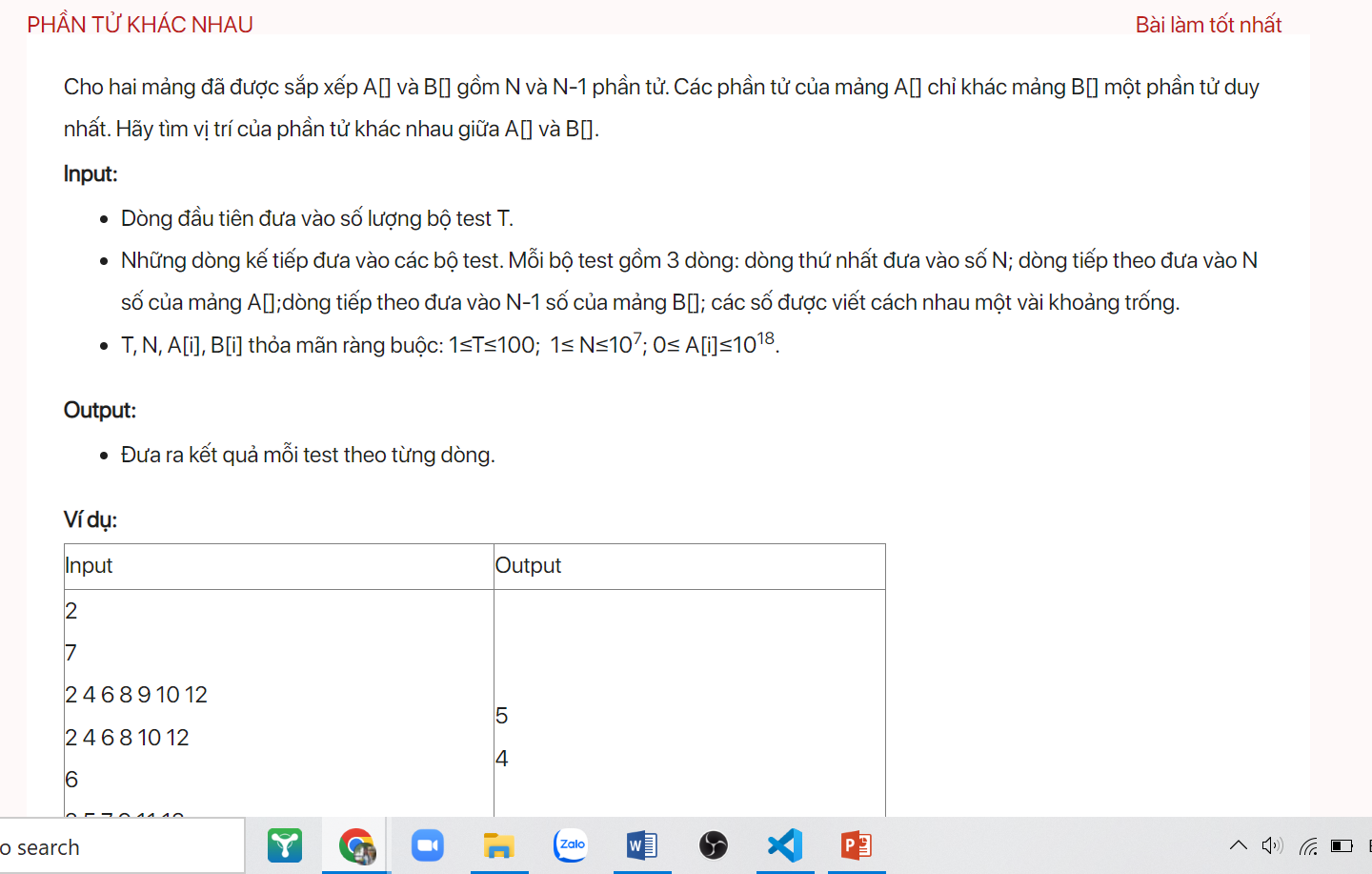
        }

        sort(c,c+n+m);

        cout<<c[k-1]<<endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin>>t;

    while(t--){

        int n;

        cin>>n;

        int a[n];

        int b[n-1];

        for(int i=0;i<n;i++){

            cin>>a[i];

        }

        for(int i=0;i<n-1;i++){

            cin>>b[i];

        }

        for(int i=0;i<n-1;i++){

            if(a[i]!=b[i]){

                cout<<i+1;

                break;

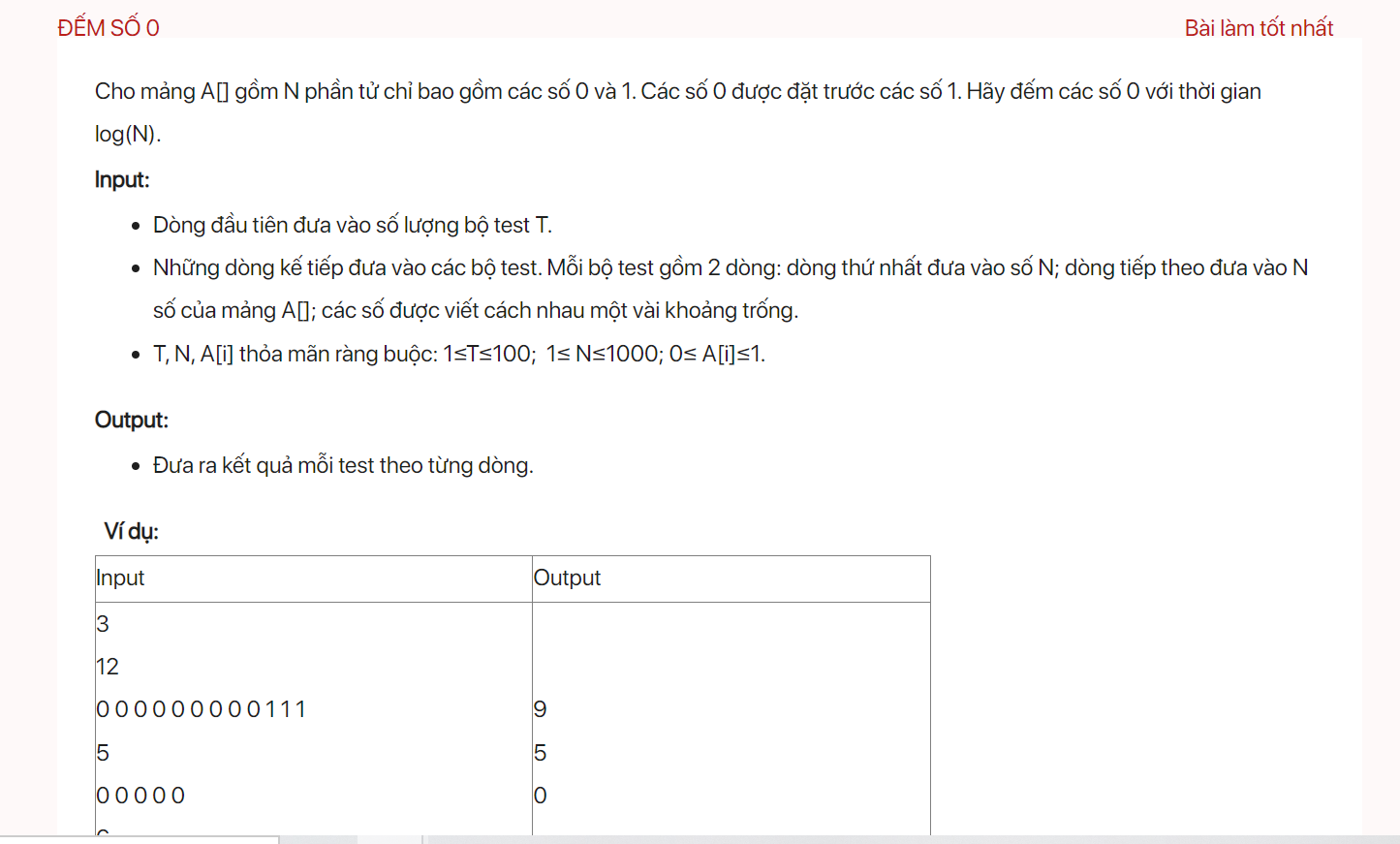
            }

        }

        cout<<endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

int main(){

    int t;

    cin>>t;

    while(t--){

        int n;

        cin>>n;

        int a[n];

        int cnt=0;

        for(int i=0;i<n;i++){

            cin>>a[i];

            if(a[i]==0)cnt++;

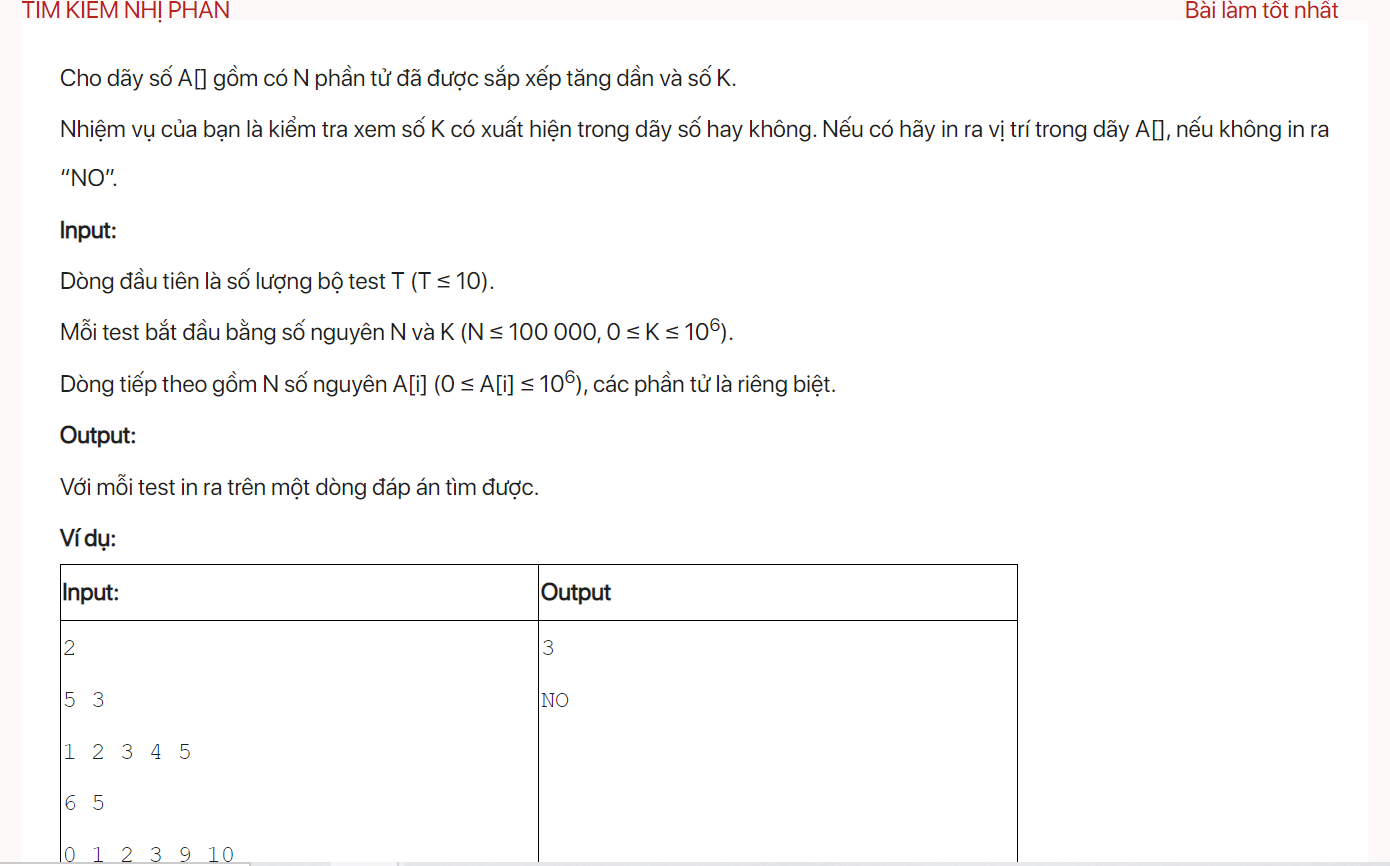
        }

        cout<<cnt;

        cout<<endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

void binarysearch(int l,int r,int k,int a[]){

    int tam=0;

    while(l<=r){

        int m=(l+r)/2;

        if(k==a[m]){

            cout<<m+1;

            tam=1;

            break;

        }

        else if(k<a[m]){

            r=m-1;

        }

        else if(k>a[m]){

            l=m+1;

        }

    }

    if(tam==0){

        cout<<"NO";

    }

}

int main(){

    int t;

    cin>>t;

    while(t--){

        int n,k;

        cin>>n>>k;

        int a[n];

        for(int i=0;i<n;i++){

            cin>>a[i];

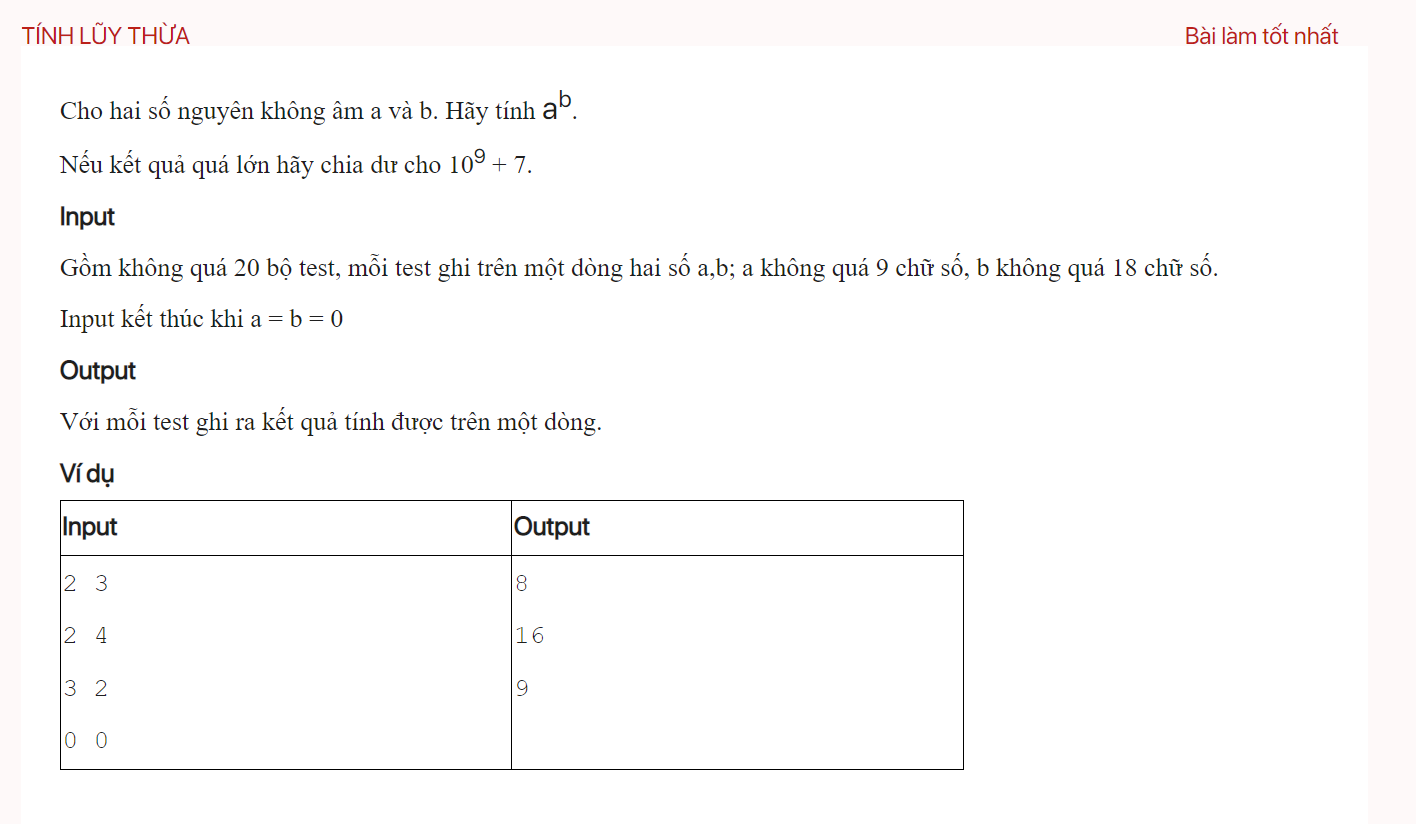
        }

        binarysearch(0,n-1,k,a);

        cout<<endl;

    }

}



#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll c=pow(10,9)+7;

ll binpow(ll a,ll b ){

    if(b==0)return 1;

    ll x=binpow(a,b/2);

    if(b%2==1){

        return ((((x%c)\*(x%c))%c)\*(a%c))%c;

    }

    else{

        return ((x%c)\*(x%c))%c;

    }

}

int main(){

    ll a,b;

   while(cin>>a&&cin>>b){

    if(a==0&&b==0)break;

    cout<<binpow(a,b)<<endl;

   }

}