#include<bits/stdc++.h>

#define INF 0x3f3f3f3f //1061109567

#define llINF 9223372036854775807

#define pi 3.141592653589793

#define ll long long

**using** **namespace** std;

**const** ll maxn=1e3+7;

**const** **double** eps=1e-10;

**const** ll mod=1e9+7;

#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0);

//注意到n的最大数据为1e9，因此用数组存储所有层数的情况是不可能的，即使我们记录的是各层到s层的距离也需要5e8

//但是这道题同样有一个有趣的数据，k的最大值是min(n-1,1000)，也就是k<=1000，关闭的厕所最多只有1000个

//那么实际上我们最多只需要移动1000层（如果起始位置在1层或者n层你只有单向可以走所以不是500层）就肯定能找到一层未关闭的厕所，因而我们实际只需要记录距离s层在500层以内厕所的开关情况即可

**int** main()

{

    ios::sync\_with\_stdio(**false**);

    ll t,n,s,k;

**int** flag[1010];

    cin>>t;

**while**(t--)

    {

        memset(flag,0,**sizeof**(flag));

        cin>>n>>s>>k;

        ll x;

**for**(ll i=0;i<k;i++)

        {

            cin>>x;

**if**(abs(s-x)<1010) flag[abs(s-x)]++;//flag[i]记录了距离s层i层的层数中有几层关闭了

        }

**if**(flag[0]==0) cout<<0<<endl;//s层未关闭直接输出0

**else**

        {

**for**(ll i=1;i<1010;i++)

            {

**if**(s-i>0&&s+i<=n)    //这里判断距离s层i层的两个层数是否都在1--n层这个范围内

                {

**if**(flag[i]<2)

                    {

                        cout<<i<<endl;

**break**;

                    }

                }

**else** **if**(flag[i]<1)

                {

                    cout<<i<<endl;

**break**;

                }

            }

        }

    }

}