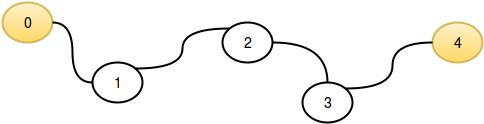
Problem: Flatland Space Station

Flatland is a country with  cities,  of which have space stations. Each city, , is numbered with a distinct index from  to , and each city  is connected to city  by a bidirectional road that is  in length.

For example, if  and cities  and  have space stations, then Flatland looks like this:



For each city, determine its distance to the *nearest* space station and *print the maximum* of these distances.

**Input Format**

The first line consists of two space-separated integers,  and  .   
The second line contains  space-separated integers describing the respective indices of each city having a space-station. These values are *unordered* and unique.

**Constraints**

* It is guaranteed that there will be at least  city with a space station, and no city has more than one.

**Output Format**

Print an integer denoting the maximum distance that an astronaut in a Flatland city would need to travel to reach the nearest space station.

**Sample Input 0**

5 2

0 4

**Sample Output 0**

2

**Explanation 0**

This sample corresponds to the example given in the problem statement above. The distance to the nearest space station for each city is listed below:

* has distance , as it contains a space station.
* has distance  to the space station in .
* has distance  to the space stations in  and .
* has distance  to the space station in .
* has distance , as it contains a space station.

We then take , and print  as our answer.

**Sample Input 1**

6 6

0 1 2 4 3 5

**Sample Output 1**

0

**Explanation 1**

In this sample,  so every city has space station and we print  as our answer.

Solution

int main()

{

long cities, spaceStations;

cin>>cities >>spaceStations;

long cwss[spaceStations]; //cities with space stations

for(long i=0; i<spaceStations; i++) //takes the values of spaceStation cities

{

cin>>cwss[i];

}

/\*processing the citites\*/

long max=0;

for(long i=0; i<cities; i++)

{

long temp=2147483647;

for(long j=0; j<spaceStations; j++)

{

if( abs(i-cwss[j])<temp )

{temp=abs(i-cwss[j]);}

}

if(temp>max)

{max=temp;}

}

cout<<max;

return 0;

}

`’Anshul AgGarwal