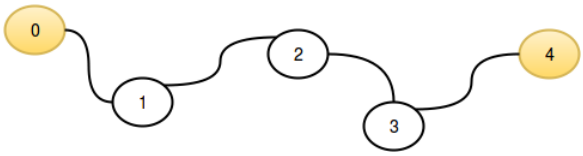
Flatland Space Stations

by Shafaet

Problem Submissions Leaderboard Discussions Editorial

Flatland is a country with n cities, m of which have space stations. Each city, c_i, is numbered with a distinct index from 0 to $n-1$, and each city c_i is connected to city c_{i+1} by a bidirectional road that is $1\ km$ in length.

For example, if $n = 5$ and cities c_0 and c_4 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, n and m .
The second line contains m space-separated integers describing the respective indices of each city having a space-station. These values are *unordered* and unique.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq n$
- It is guaranteed that there will be at least 1 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:

- c_0 has distance $0\ km$, as it contains a space station.
- c_1 has distance $1\ km$ to the space station in c_0.
- c_2 has distance $2\ km$ to the space stations in c_0 and c_4.
- c_3 has distance $1\ km$ to the space station in c_4.
- c_4 has distance $0\ km$, as it contains a space station.

We then take $max(0, 1, 2, 1, 0) = 2$, and print 2 as our answer.

Sample Input 1

```
6 6
0 1 2 4 3 5
```

Sample Output 1

```
0
```

Explanation 1

In this sample, $n = m$ so every city has space station and we print 0 as our answer.

Submissions: 6479
Max Score: 25
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

    static void Main(String[] args) {
        string[] tokens_n = Console.ReadLine().Split(' ');
        int n = Convert.ToInt32(tokens_n[0]);
        int m = Convert.ToInt32(tokens_n[1]);
        string[] c_temp = Console.ReadLine().Split(' ');
        int[] c = Array.ConvertAll(c_temp,Int32.Parse);



           int ans = 0;
            Array.Sort(c);

            int primero = c[0] - 0;
            int ultimo = (n - 1) - c[c.Length - 1];

            ans = Math.Max(primero, ultimo);

            for (int i = 0; i + 1 < c.Length; i++)
            {
                int prom = (c[i + 1] + c[i]) / 2;
                int min = Math.Min(prom - c[i], c[i + 1] - prom);
                ans = Math.Max(ans, min);
            }

            Console.WriteLine(ans);




    }
}

```

Line: 17 Col: 12

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14
- ✔ Test Case #15
- ✔ Test Case #16
- ✔ Test Case #17
- ✔ Test Case #18
- ✔ Test Case #19

Next Challenge