**Safe Houses**

Your first project in the Academy is to develop a simple game about a hero who must save the princess and rule the kingdom. However, as in all great games, the first few levels are difficult and the evil uncle's guards are constantly on the hunt for our hero. Luckily, there are several safehouses, where the hero can hide.

When starting a game, safehouses spawn at random locations and, according to the difficulty level, our hero may spawn far from or near to a safehouse. Your task is to determine the spawning location of the hero in *Easy Mode*. To do that, you must find the greatest distance from a possible spawn location to a safehouse and not spawn there.

In the example below, there are 6 possible spawn points, 1 of them has a safehouse (S), and 5 don't (X). The greatest distance from X to S is therefore 3. This is the spawn point that must be excluded in easy mode.

X - X - X - S - X - X

3 2 1 0 1 2

If there are multiple safehouses, for each spawn location you need to determine the nearest safe house, and then return the largest of those distances. (Check the second example)

**Input**

Exactly two lines:

* The number of spawn points
* Numbers (**unique, unordered**) - the indices (**zero-based**) of spawn points with safe houses.

**Output**

Exactly one line:

* The greatest distance

**Constraints**

* 2 < spawn points < 30
* there will always be at least one safehouse

**Sample tests**

**Input**

6

3

**Output**

3

*Explanation*:

6 spawn points, spawn point 3 has a safehouse (greatest distance: 3)

x-x-x-S-x-x

3 2 1 0 1 2

**Input**

5

4 0

**Output**

2

*Explanation*:

5 spawn points, spawn points 4 and 0 have safehouses (greatest distance: 2)

S-x-x-x-S

0 1 2 1 0

**Input**

3

0 1 2

**Output**

0

*Explanation*:

3 spawn points, spawn points 0, 1 and 2 have safehouses (greatest distance: 0)

S-S-S

0 0 0