Chief's bot is playing an old DOS based game. There is a row of buildings of different heights arranged at each index along a number line. The bot starts at building 0 and at a height of 0. You must determine the minimum energy his bot needs at the start so that he can jump to the top of each building without his energy going below zero.

Units of height relate directly to units of energy. The bot's energy level is calculated as follows:

- If the bot's $botEnergy$ is less than the height of the building, his $newEnergy = botEnergy - (height - botEnergy)$
- If the bot's $botEnergy$ is greater than the height of the building, his $newEnergy = botEnergy + (botEnergy - height)$

For example, building heights are given as $h = [2, 3, 4, 3, 2]$. If the bot starts with $botEnergy = 4$, we get the following table:

```
botEnergy   height  delta
4               2       +2
6               3       +3
9               4       +5
14              3       +11
25              2       +23
48
```

That allows the bot to complete the course, but may not be the minimum starting value. The minimum starting $botEnergy$ in this case is 3.

Function Description

Complete the *chiefHopper* function in the editor below. It should return an integer that represents the minimum starting $botEnergy$ that will allow completion of the course.

chiefHopper has the following parameter(s):

- *arr*: an array of integers that represent building heights

Input Format

The first line contains an integer n, the number of buildings.

The next line contains n space separated integers $h[1] \ldots h[n]$ representing the heights of the buildings.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq h[i] \leq 10^5, i \in [1, n]$

Output Format

Print a single integer representing minimum units of energy required to complete the game.

Sample Input 0

```
5
3 4 3 2 4
```

Sample Output 0

```
4
```

Explanation 0

If initial energy is 4, after step 1 energy is 5, after step 2 it's 6, after step 3 it's 9 and after step 4 it's 16, finally at step 5 it's 28.
If initial energy were 3 or less, the bot could not complete the course.

Sample Input 1

```
3
4 4 4
```

Sample Output 1

```
4
```

Explanation 1

In the second test case if bot has energy 4, it's energy is changed by (4 - 4 = 0) at every step and remains 4.

Sample Input 2

```
3
1 6 4
```

Sample Output 2

```
3
```

Explanation 2

```
botEnergy   height  delta
3           1       +2
5           6       -1
4           4       0
4
```

We can try lower values to assure that they won't work.