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 $newEnergy = botEnergy + (botEnergy - height)$

Example

$arr = [2, 3, 4, 3, 2]$

Starting 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.

chiefHopper has the following parameter(s):

- int arr[n]: building heights

Returns

- int: the minimum starting $botEnergy$

Input Format

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

The next line contains n space-separated integers $arr[1]..arr[n]$, the heights of the buildings.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 10^5$ where $1 \leq i \leq n$

Sample Input

Change Theme Language Python 3

```python
#!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'chiefHopper' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER_ARRAY arr as parameter.
#

def chiefHopper(arr):
    # Write your code here
    BE=0
    for height in reversed(arr):
        BE = math.ceil((BE+height)/2)

    return BE


```

Line: 35 Col: 5

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

Congratulations

You solved this challenge. Would you like to challenge your friends?

Test case 0
Test case 1
Test case 2

Compiler Message

Success

Input (stdin) Download

```
5
```