

Problema 1

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

Tenemos n torres formadas por bloques, numeradas desde el 1 hasta n . La i -ésima torre consiste de a_i bloques. Por ejemplo si tenemos el arreglo $a = [1\ 5\ 10\ 2]$, significa que la tercera torre ($i = 3$), esta formada por 10 bloques ($a_3 = 10$) apilados el uno sobre el otro, del mismo modo la primera torre ($i = 1$) tiene solo un bloque ($a_1 = 1$).

En un movimiento, puedes mover un bloque desde la torre i , a la torre j , si y solo si la cantidad de bloques en la torre i es estrictamente mayor que la cantidad de bloques en la torre j , es decir $a_i > a_j$. Este movimiento incrementa a_j en 1 y disminuye a_i en 1. Puedes realizar tantos movimientos como quieras (posiblemente cero).

Entradas

1. n : la cantidad de torres.
2. a : Un arreglo con la cantidad de bloques de cada torre desde 1 hasta n .

Salidas

La cantidad más grande de bloques que puedes tener en el bloque 1 después de realizar cualquier número de movimientos.

Input

Test 1: $n=5$ y $a=[1\ 1\ 1\ 1\ 1]$
Test 2: $n=3$ y $a=[10\ 9\ 8]$
Test 3: $n=3$ y $a=[1\ 2\ 3]$
Test 4: $n=3$ y $a=[1\ 2\ 2]$
Test 5: $n=2$ y $a=[1\ 1000000000]$
Test 6: $n=10$ y $a=[3\ 8\ 6\ 7\ 4\ 1\ 2\ 4\ 10\ 1]$

Output

Test 1: 1
Test 2: 10
Test 3: 3
Test 4: 2
Test 5: 5000000001
Test 6: 9

Note

La prueba 1 retorna 1 porque todas las otras torres tienen menor o igual cantidad de bloques, por lo que no podemos añadir ninguno a la primera torre.

La prueba 3 retorna 3 ya que movemos un bloque de la torre 2 a la torre 1, dandonos como resultado $[2\ 1\ 3]$, luego movemos un bloque de la torre 3 a la torre 1 y nos da $[3\ 1\ 2]$. No podemos añadir más bloques porque $2 < 3$.