Combine

SEASON 8 – SIXTH ROUND

Viktor loves jellies.

Today he bought **N** jellies and put them on a table in such a way that jelly number **i** has size equal to **a[i]** and it is adjacent to jellies with numbers **i+1** and **i-1** (if they exist).

We can preform the following operation multiple times:

Choose **two adjacent jellies of equal sizes** and combine them into one with size equal to the **sum of the two previous sizes**. The new jelly will be adjacent two the neighbours of the jellies the operation was applied to, i.e. if the operation was applied to jellies **p** and **p+1**, the new jelly will be adjacent to jellies at positions **p-1** и **p+2** (if they exist).

After preforming this operation multiple times, we will end up with a configuration of jellies. A **beuty** of a configuration is the **minimal size of a jelly in it**. Help Viktor by writing a program that computes the **maximal possible beauty of a configuration of jellies** that he can get.

Input

The first line of the file combine.in contains **N** – the initial number of jellies that Viktor bought. The second line contains the initial sizes of the jellies – **a[1], a[2], ..., a[N]**.

Output

The output file combine.out must contain one line with the **maximal possible** beauty of a configuration of jellies.

Constraints:

1 ≤ N ≤ 1000

1 ≤ a[i] ≤ 10^9

Time limit: 1 sec

Memory limit: 256 MB

Example test:

Input (combine.in)	Output (combine.out)
4 1 1 1 1	4