

Problem K. Process with Constant Sum

Input file: *standard input*
Output file: *standard output*
Time limit: 2.5 seconds
Memory limit: 512 mebibytes

Suppose we have an array a whose elements are non-negative integers. We can apply operations of two types to this array:

1. $a_i \leftarrow a_i - 2, a_{i+1} \leftarrow a_{i+1} + 2$.
2. If $a_{i+1} = 0$: $a_i \leftarrow a_i - 1, a_{i+2} \leftarrow a_{i+2} + 1$.

After any operation, all the elements must remain non-negative.

It is easy to see that we cannot do such operations indefinitely. We'll call a chain of operations *maximal* if no valid operation can occur after performing this chain of operations.

Now you are given an array b of length n . You should handle two types of queries:

- 1 p x — assignment $b_p \leftarrow x$.
- 2 l r — let us treat the subsegment $b[l..r]$ as array a . You have to find the maximum number of zeroes which can be in that array after applying some *maximal* chain of operations. Note that we don't really change the array b during this query.

Input

The first line of input contains one integer n , the length of array b ($1 \leq n \leq 2 \cdot 10^5$).

The second line contains n integers separated by spaces: the initial elements of array b ($0 \leq b_i \leq 10^5$).

The third line contains one integer q , the number of queries ($1 \leq q \leq 2 \cdot 10^5$).

Each of the next q lines contains a query.

For queries of the first type, $1 \leq p \leq n$ and $0 \leq x \leq 10^5$.

For queries of the second type, $1 \leq l \leq r \leq n$.

Output

For each query of the second type, print the answer on a separate line. Answers should be printed in the same order as queries.

Examples

standard input	standard output
3 2 2 2 9 2 1 3 1 1 1 2 1 3 1 2 1 2 1 3 1 1 4 2 1 3 2 1 2 2 2 3	2 2 0 1 1 0
1 20500 1 2 1 1	0