D. k-Maximum Subsequence Sum

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Consider integer sequence $a_1, a_2, ..., a_n$. You should run queries of two types:

- The query format is "$0\ i\ val$". In reply to this query you should make the following assignment: $a_i = val$.
- The query format is "$1\ l\ r\ k$". In reply to this query you should print the maximum sum of at most k non-intersecting subsegments of sequence $a_l, a_{l+1}, ..., a_r$. Formally, you should choose at most k pairs of integers $(x_1, y_1), (x_2, y_2), ..., (x_t, y_t)$ $(l \le x_1 \le y_1 < x_2 \le y_2 < ... < x_t \le y_t \le r;\ t \le k)$ such that the sum $a_{x_1} + a_{x_1+1} + ... + a_{y_1} + a_{x_2} + a_{x_2+1} + ... + a_{y_2} + ... + a_{x_t} + a_{x_t+1} + ... + a_{y_t}$ is as large as possible. Note that you should choose at most k subsegments. Particularly, you can choose 0 subsegments. In this case the described sum considered equal to zero.

Input

The first line contains integer n $(1 \le n \le 10^5)$, showing how many numbers the sequence has. The next line contains n integers $a_1, a_2, ..., a_n$ $(|a_i| \le 500)$.

The third line contains integer m $(1 \le m \le 10^5)$ — the number of queries. The next m lines contain the queries in the format, given in the statement.

All changing queries fit into limits: $1 \le i \le n,\ |val| \le 500$.

All queries to count the maximum sum of at most k non-intersecting subsegments fit into limits: $1 \le l \le r \le n$, $1 \le k \le 20$. It is guaranteed that the number of the queries to count the maximum sum of at most k non-intersecting subsegments doesn't exceed 10000.

Output

For each query to count the maximum sum of at most k non-intersecting subsegments print the reply — the maximum sum. Print the answers to the queries in the order, in which the queries follow in the input.

Examples

input

```
9
9 -8 9 -1 -1 -1 9 -8 9
3
1 1 9 1
1 1 9 2
1 4 6 3
```

output

```
17
25
0
```

input

```
15
-4 8 -3 -10 10 4 -7 -7 0 -6 3 8 -10 7 2
15
1 3 9 2
1 6 12 1
0 6 5
0 10 -7
1 4 9 1
1 7 9 1
```

```
0 10 -3
1 4 10 2
1 3 13 2
1 4 11 2
0 15 -9
0 13 -9
0 11 -10
1 5 14 2
1 6 12 1
```

output

```
14
11
15
0
15
26
18
23
8
```

Note

In the first query of the first example you can select a single pair $(1, 9)$. So the described sum will be 17.

Look at the second query of the first example. How to choose two subsegments? (1, 3) and (7, 9)? Definitely not, the sum we could get from (1, 3) and (7, 9) is 20, against the optimal configuration (1, 7) and (9, 9) with 25.

The answer to the third query is 0, we prefer select nothing if all of the numbers in the given interval are negative.