A. Rectangle Puzzle

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two rectangles on a plane. The centers of both rectangles are located in the origin of coordinates (meaning the center of the rectangle's symmetry). The first rectangle's sides are parallel to the coordinate axes: the length of the side that is parallel to the Ox axis, equals w, the length of the side that is parallel to the Oy axis, equals h. The second rectangle can be obtained by rotating the first rectangle relative to the origin of coordinates by angle α.

Your task is to find the area of the region which belongs to both given rectangles. This region is shaded in the picture.

Input

The first line contains three integers w, h, α $(1 \le w, h \le 10^6; 0 \le \alpha \le 180)$. Angle α is given in degrees.

Output

In a single line print a real number — the area of the region which belongs to both given rectangles.

The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Examples

input
```
1 1 45
```
output
```
0.828427125
```

input
```
6 4 30
```
output
```
19.668384925
```

Note

The second sample has been drawn on the picture above.

B. Maximum Xor Secondary

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bike loves looking for the second maximum element in the sequence. The second maximum element in the sequence of distinct numbers $x_1, x_2, ..., x_k$ $(k > 1)$ is such maximum element x_j, that the following inequality holds: $x_j \neq \max_{i=1}^{k} x_i$.

The lucky number of the sequence of distinct positive integers $x_1, x_2, ..., x_k$ $(k > 1)$ is the number that is equal to the bitwise excluding OR of the maximum element of the sequence and the second maximum element of the sequence.

You've got a sequence of distinct positive integers $s_1, s_2, ..., s_n$ $(n > 1)$. Let's denote sequence $s_l, s_{l+1}, ..., s_r$ as $s[l..r]$ $(1 \le l < r \le n)$. Your task is to find the maximum number among all lucky numbers of sequences $s[l..r]$.

Note that as all numbers in sequence s are distinct, all the given definitions make sence.

Input

The first line contains integer n $(1 < n \le 10^5)$. The second line contains n distinct integers $s_1, s_2, ..., s_n$ $(1 \le s_i \le 10^9)$.

Output

Print a single integer — the maximum lucky number among all lucky numbers of sequences $s[l..r]$.

Examples

input

```
5
5 2 1 4 3
```

output

```
7
```

input

```
5
9 8 3 5 7
```

output

```
15
```

Note

For the first sample you can choose $s[4..5] = \{4, 3\}$ and its lucky number is $(4\ xor\ 3) = 7$. You can also choose $s[1..2]$.

For the second sample you must choose $s[2..5] = \{8, 3, 5, 7\}$.

C. Game on Tree

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Momiji has got a rooted tree, consisting of n nodes. The tree nodes are numbered by integers from 1 to n. The root has number 1. Momiji decided to play a game on this tree.

The game consists of several steps. On each step, Momiji chooses one of the remaining tree nodes (let's denote it by v) and removes all the subtree nodes with the root in node v from the tree. Node v gets deleted as well. The game finishes when the tree has no nodes left. In other words, the game finishes after the step that chooses the node number 1.

Each time Momiji chooses a new node uniformly among all the remaining nodes. Your task is to find the expectation of the number of steps in the described game.

Input

The first line contains integer n $(1 \le n \le 10^5)$ — the number of nodes in the tree. The next $n - 1$ lines contain the tree edges. The i-th line contains integers a_i, b_i $(1 \le a_i, b_i \le n;\ a_i \ne b_i)$ — the numbers of the nodes that are connected by the i-th edge.

It is guaranteed that the given graph is a tree.

Output

Print a single real number — the expectation of the number of steps in the described game.

The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Examples

input

```
2
1 2
```

output

```
1.50000000000000000000
```

input

```
3
1 2
1 3
```

output

```
2.00000000000000000000
```

Note

In the first sample, there are two cases. One is directly remove the root and another is remove the root after one step. Thus the expected steps are:

$$1 \times (1 / 2) + 2 \times (1 / 2) = 1.5$$

In the second sample, things get more complex. There are two cases that reduce to the first sample, and one case cleaned at once. Thus the expected steps are:

$$1 \times (1 / 3) + (1 + 1.5) \times (2 / 3) = (1 / 3) + (5 / 3) = 2$$

D. k-Maximum Subsequence Sum

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Consider integer sequence $a_1, a_2, \ldots, a_n$. 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 $a_l, a_{l+1}, \ldots, a_r$. Formally, you should choose at most k pairs of integers $(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)$ $(l \le x_1 \le y_1 < x_2 \le y_2 < \ldots < x_t \le y_t \le r;\ t \le k)$ such that the sum $a_{x_1} + a_{x_1+1} + \ldots + a_{y_1} + a_{x_2} + a_{x_2+1} + \ldots + a_{y_2} + \ldots + a_{x_t} + a_{x_t+1} + \ldots + a_{y_t}$ 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 $a_1, a_2, \ldots, a_n$ $(|a_i| \le 500)$.

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.

E. Sequence Transformation

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You've got a non-decreasing sequence $x_1, x_2, \ldots, x_n$ $(1 \le x_1 \le x_2 \le \ldots \le x_n \le q)$. You've also got two integers a and b $(a \le b;\ a \cdot (n - 1) < q)$.

Your task is to transform sequence $x_1, x_2, \ldots, x_n$ into some sequence $y_1, y_2, \ldots, y_n$ $(1 \le y_i \le q;\ a \le y_{i+1} - y_i \le b)$. The transformation price is the following sum: $\sum_{i=1}^{n} (y_i - x_i)^2$. Your task is to choose such sequence y that minimizes the described transformation price.

Input

The first line contains four integers n, q, a, b $(2 \le n \le 6000;\ 1 \le q, a, b \le 10^9;\ a \cdot (n - 1) < q;\ a \le b)$.

The second line contains a non-decreasing integer sequence $x_1, x_2, \ldots, x_n$ $(1 \le x_1 \le x_2 \le \ldots \le x_n \le q)$.

Output

In the first line print n real numbers — the sought sequence $y_1, y_2, \ldots, y_n$ $(1 \le y_i \le q;\ a \le y_{i+1} - y_i \le b)$. In the second line print the minimum transformation price, that is, $\sum_{i=1}^{n} (y_i - x_i)^2$.

If there are multiple optimal answers you can print any of them.

The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Examples

input

```
3 6 2 2
1 4 6
```

output

```
1.666667 3.666667 5.666667
0.666667
```

input

```
10 100000 8714 9344
3378 14705 17588 22672 32405 34309 37446 51327 81228 94982
```

output

```
1.000000 8715.000000 17429.000000 26143.000000 34857.000000 43571.000000 52285.000000 61629.000000 70973.000000
80317.000000
797708674.000000
```