Each exercise is 2 points. There are 4 exercise including 4.2.1

4.1 The Most Frequent Symbol

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

For any string S you can easily find the most frequent symbol. What about queries for some substring of S? Can you find the most frequent symbol quickly?

You are given a string $S = s_1 s_2 \dots s_{|S|}$ and a set of queries, where each query is a pair of indices (l, r), $1 \le l \le r \le |S|$. For each query (l, r), find the most frequent symbol in $s_l s_{l+1} \dots s_r$.

Input

The first line contains a strings S $(1 \le |S| \le 50\,000)$. The string S consists of small Latin letters only.

The second line contains an integer Q $(1 \le Q \le 50\,000)$, the number of queries.

Each of the next Q lines contains two integers l and r $(1 \le l \le r \le |S|)$, positions of the first and the last symbols in the substring.

Output

For each query, print a single line with the most frequent symbol in the substring. In case of multiple most frequent symbols output any of them.

Examples

standard input	standard output
abacaba 3 1 1 1 7 2 4	a a c
abba 6 1 1 2 2 1 2 2 3 1 1 2 4	a b b b a b

4.2 Maximal Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Find a pair of points on the line with maximum distance between them:

$$\text{Distance}(a, b) = |x_a - x_b|.$$

To make this problem more interesting, let's find such a pair after each step of adding points to the line.

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of points on the line.

The i-th of the next n lines contains coordinates x_i of the i-th point ($0 \le x_i \le 10^9$ is integer).

All points are different.

Output

Print n lines. The i-th line should contain two numbers f_i and s_i ($1 \le f_i, s_i \le i$), the indices of two points with maximum distance among the first i points.

In case of multiple correct pairs of points, print any of them.

Examples

standard input	standard output
3 1 2 3	1 1 1 2 1 3
5 3 2 1 50 49	1 1 1 2 1 3 3 4 3 4

4.2.1 is another exercise

4.2.1 Manhattan Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Find a pair of points on the Cartesian plane with maximum Manhattan distance between them:

$$\text{ManhattanDistance}(a, b) = |a_x - b_x| + |a_y - b_y|.$$

To make this problem more interesting, let's find such a pair after each step of adding points to the plane.

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of points on the plane.

The i-th of the next n lines contains coordinates x_i, y_i of the i-th point ($0 \le x_i, y_i \le 10^9$ are integers). All points are different.

Output

Print n lines. The i-th line should contain two numbers f_i and s_i ($1 \le f_i, s_i \le i$), the indices of two points with maximum Manhattan distance among the first i points.

In case of multiple correct pairs of points, print any of them.

Examples

standard input	standard output
3 1 1 2 1 1 3	1 1 2 1 2 3
5 2 2 1 3 1 1 3 1 3 3	1 1 1 2 3 2 4 2 4 2

4.4 Maximal Sum Subarray

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Let $A[1..n]$ be an array of integers. For all i from 1 to n find a subarray with maximum sum that covers the position i (more formally, for every i, find the largest value $A[l] + A[l+1] + \cdots + A[r]$ among all pairs of indices l and r such that $1 \le l \le i \le r \le n$).

Input

The first line contains an integer n ($1 \le n \le 100\,000$), the number of elements in A.

The second line contains integers $A[1], A[2], \ldots, A[n]$ ($-10^6 \le A[i] \le 10^6$).

Output

Print n integers separated by spaces. The i-th of them should be equal to the maximal sum of subarray among all that cover the position i in A.

Examples

standard input	standard output
3 -1000000 1000000 -1000000	0 1000000 0
4 1 2 3 4	10 10 10 10
5 2 -3 -3 -3 2	2 -1 -3 -1 2