

Problem A. Min queries

Input file: qmin.inp
Output file: qmin.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

Cho một mảng a gồm n phần tử và q truy vấn.

Mỗi truy vấn có dạng “ $l\ r$ ” và bạn cần in ra giá trị của số nhỏ nhất trong các số $a[l], a[l+1], \dots, a[r]$.

Input

Dòng đầu tiên chứa số n ($1 \leq n \leq 100\,000$).

Dòng thứ 2 chứa n số a_1, a_2, \dots, a_n là các phần tử của mảng ($0 \leq a_i \leq 10^9$).

Dòng thứ 3 chứa số q ($1 \leq q \leq 100\,000$).

q dòng tiếp theo, mỗi dòng chứa 2 số l và r ($1 \leq l \leq r \leq n$).

Output

Với mỗi truy vấn, in ra đáp án của truy vấn đó trên một dòng.

Examples

qmin.inp	qmin.out
10	3
6 20 7 14 12 11 15 20 3 5	6
10	20
4 9	3
1 3	3
8 8	3
8 10	15
4 10	7
4 10	3
7 7	11
3 8	
1 9	
6 8	

Problem B. Min queries 2

Input file: qmin2.inp
Output file: qmin2.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

Cho một mảng a gồm n phần tử và q truy vấn.

Mỗi truy vấn có dạng “ $l\ r$ ” và bạn cần in ra giá trị của số nhỏ thứ nhì trong các số $a[l], a[l + 1], \dots, a[r]$.

Input

Dòng đầu tiên chứa số n ($2 \leq n \leq 100\,000$).

Dòng thứ 2 chứa n số a_1, a_2, \dots, a_n là các phần tử của mảng ($0 \leq a_i \leq 10^9$).

Dòng thứ 3 chứa số q ($1 \leq q \leq 100\,000$).

q dòng tiếp theo, mỗi dòng chứa 2 số l và r ($1 \leq l < r \leq n$).

Output

Với mỗi truy vấn, in ra đáp án của truy vấn đó trên một dòng.

Examples

qmin2.inp	qmin2.out
10	3
3 2 5 0 9 3 8 3 4 5	3
10	5
5 8	5
6 9	2
2 3	2
9 10	3
2 8	3
2 6	4
4 8	3
4 9	
8 9	
3 6	

Problem C. Multiplication query

Input file: `qmul.inp`
Output file: `qmul.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Cho một mảng a gồm n phần tử và q truy vấn.

Mỗi truy vấn có dạng " $l\ r$ " và bạn cần in ra tích của các số $a[l], a[l+1], \dots, a[r]$ modulo 1 000 000 007.

Input

Dòng đầu tiên chứa số n ($1 \leq n \leq 100\,000$).

Dòng thứ 2 chứa n số a_1, a_2, \dots, a_n là các phần tử của mảng ($1 \leq a_i \leq 10^9$).

Dòng thứ 3 chứa số q ($1 \leq q \leq 100\,000$).

q dòng tiếp theo, mỗi dòng chứa 2 số l và r ($1 \leq l \leq r \leq n$).

Output

Với mỗi truy vấn, in ra đáp án của truy vấn đó trên một dòng.

Examples

<code>qmul.inp</code>	<code>qmul.out</code>
10	80
5 4 9 8 8 10 6 7 4 9	252
10	20
5 6	107520
8 10	180
1 2	80
4 9	34836480
1 3	3870720
5 6	288
2 10	9
2 9	
2 4	
10 10	

Problem D. Bracket

Input file: `bracket.inp`
Output file: `bracket.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Cho một xâu s chỉ gồm 2 loại kí tự “(” và “)”.

Một xâu chỉ gồm 2 loại kí tự trên được gọi là tốt khi mà tồn tại cách chèn dấu $+$ và số 1 vào xâu, ta có được một biểu thức có nghĩa. Ví dụ, “()((()” là một xâu tốt vì ta có thể chèn $+$ và 1 để tạo ra $(1 + 1)(1 + (1 + 1))$ là một biểu thức có nghĩa; trong khi “(” không phải là một xâu tốt.

Ta quan tâm đến độ dài của xâu con tốt dài nhất của s , đồng thời muốn biết có bao nhiêu xâu đạt được độ dài đó.

Input

Xâu s chỉ gồm hai loại kí tự “(” và “)” ($|s| \leq 10^5$).

Output

Độ dài của xâu con đúng dài nhất, và số xâu con như thế.

Nếu s không có xâu con đúng, in “0 1”.

Examples

<code>bracket.inp</code>	<code>bracket.out</code>
<code>()()((()</code>	<code>2 3</code>

Problem E. Greatest common divisor queries

Input file: `gcds.inp`
Output file: `gcds.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Cho một mảng a gồm n phần tử và q truy vấn.

Mỗi truy vấn có dạng “ x ” và bạn cần đếm số đoạn con liên tiếp của mảng mà ước chung lớn nhất của đoạn đó bằng x .

Input

Dòng đầu tiên chứa số n ($1 \leq n \leq 100\,000$).

Dòng thứ 2 chứa n số a_1, a_2, \dots, a_n là các phần tử của mảng ($1 \leq a_i \leq 10^9$).

Dòng thứ 3 chứa số q ($1 \leq q \leq 100\,000$).

q dòng tiếp theo, mỗi dòng chứa 1 số x ($1 \leq x \leq 10^9$).

Output

Với mỗi truy vấn, in ra đáp án của truy vấn đó trên một dòng.

Examples

<code>gcds.inp</code>	<code>gcds.out</code>
10	2
7 3 10 3 8 4 3 1 4 8	4
10	3
8	1
4	0
3	4
7	3
6	0
4	1
3	0
9	
10	
5	

Explanations

Truy vấn thứ 2 có 4 đoạn con mà ước chung lớn nhất bằng 4 là $[5, 6], [6, 6], [9, 9], [9, 10]$.

Problem F. Sum queries

Input file: qsum.inp
Output file: qsum.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

Cho một mảng a gồm n phần tử và q truy vấn.

Mỗi truy vấn có dạng “ $l\ r$ ” và bạn cần tìm xem, trong các đoạn con $[x, y]$ mà $l \leq x \leq y \leq r$, đoạn nào có tổng các phần tử của đoạn lớn nhất?

Input

Dòng đầu tiên chứa số n ($1 \leq n \leq 50\,000$).

Dòng thứ 2 chứa n số a_1, a_2, \dots, a_n là các phần tử của mảng ($-40000 \leq a_i \leq 40000$).

Dòng thứ 3 chứa số q ($1 \leq q \leq 100\,000$).

q dòng tiếp theo, mỗi dòng chứa 2 số l và r ($1 \leq l \leq r \leq n$).

Output

Với mỗi truy vấn, in ra đáp án của truy vấn đó trên một dòng.

Examples

qsum.inp	qsum.out
10	35
5 18 12 -3 -19 6 -7 -10 -16 -10	30
10	-10
1 10	12
2 8	6
9 10	-7
3 7	35
4 6	30
7 9	12
1 4	35
2 10	
3 9	
1 5	

Problem G. Cake

Input file: cake.inp
Output file: cake.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

Có n người sành ăn ngồi trên một chiếc bàn dài gồm n cái bánh. n người được số lần lượt $1, 2, \dots, n$ theo thứ tự từ trái qua phải. Trước mặt mỗi người là một cái bánh, chúng cũng được đánh số như những người sành ăn.

Tại bàn, mỗi người chỉ có thể với ăn được chiếc bánh trước mặt họ, hoặc 2 cái bánh ở ngay cạnh (nếu tồn tại).

Với mỗi người, mỗi chiếc bánh có một độ thỏa mãn khác nhau. Bạn cần ghép cặp người với bánh sao cho mỗi người có đúng 1 bánh, người đó với được bánh và tổng độ thỏa mãn của tất cả người sành ăn là lớn nhất có thể.

Để làm bài khó hơn, sẽ có q truy vấn, mỗi truy vấn sẽ hỏi đáp án của 1 đoạn con của mảng.

Input

Dòng đầu tiên chứa số n ($1 \leq n \leq 100\,000$).

n dòng sau, dòng thứ i chứa 3 số nguyên không âm ($\leq 10^4$) tức độ ngon của bánh bên trái ($i-1$), bánh trước mặt (i) và bánh bên phải ($i+1$) của người i (chú ý trường hợp bánh không tồn tại).

Dòng tiếp theo là số q .

q dòng tiếp theo mỗi dòng 2 số l và r , tức là ta cần giải quyết bài toán con với người từ vị trí l đến r , ăn các bánh từ l đến r .

Output

Với mỗi truy vấn, in ra tổng độ thỏa mãn lớn nhất đạt được.

Examples

cake.inp	cake.out
3	7
0 1 2	
2 1 0	
0 3 0	
1	
1 3	