Educational Codeforces Round 15

A. Maximum Increase

1 second, 256 megabytes

You are given array consisting of n integers. Your task is to find the maximum length of an increasing subarray of the given array.

A subarray is the sequence of consecutive elements of the array. Subarray is called increasing if each element of this subarray **strictly greater** than previous.

Input

The first line contains single positive integer n ($1 \leq n \leq 10^5$) — the number of integers.

The second line contains n positive integers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq 10^9$).

Output

Print the maximum length of an increasing subarray of the given array.

input
```
5
1 7 2 11 15
```
output
```
3
```

input
```
6
100 100 100 100 100 100
```
output
```
1
```

input
```
3
1 2 3
```
output
```
3
```

B. Powers of Two

3 seconds, 256 megabytes

You are given n integers $a_1, a_2, ..., a_n$. Find the number of pairs of indexes i, j ($i < j$) that $a_i + a_j$ is a power of 2 (i. e. some integer x exists so that $a_i + a_j = 2^x$).

Input

The first line contains the single positive integer n ($1 \leq n \leq 10^5$) — the number of integers.

The second line contains n positive integers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq 10^9$).

Output

Print the number of pairs of indexes i, j ($i < j$) that $a_i + a_j$ is a power of 2.

input
```
4
7 3 2 1
```
output
```
2
```

input
```
3
1 1 1
```
output
```
3
```

In the first example the following pairs of indexes include in answer: $(1, 4)$ and $(2, 4)$.

In the second example all pairs of indexes (i, j) (where $i < j$) include in answer.

C. Cellular Network

3 seconds, 256 megabytes

You are given n points on the straight line — the positions (x-coordinates) of the cities and m points on the same line — the positions (x-coordinates) of the cellular towers. All towers work in the same way — they provide cellular network for all cities, which are located at the distance which is no more than r from this tower.

Your task is to find minimal r that each city has been provided by cellular network, i.e. for each city there is at least one cellular tower at the distance which is no more than r.

If $r = 0$ then a tower provides cellular network only for the point where it is located. One tower can provide cellular network for any number of cities, but all these cities must be at the distance which is no more than r from this tower.

Input

The first line contains two positive integers n and m ($1 \le n, m \le 10^5$) — the number of cities and the number of cellular towers.

The second line contains a sequence of n integers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$) — the coordinates of cities. It is allowed that there are any number of cities in the same point. All coordinates a_i are given in non-decreasing order.

The third line contains a sequence of m integers $b_1, b_2, ..., b_m$ ($-10^9 \le b_j \le 10^9$) — the coordinates of cellular towers. It is allowed that there are any number of towers in the same point. All coordinates b_j are given in non-decreasing order.

Output

Print minimal r so that each city will be covered by cellular network.

input
```
3 2
-2 2 4
-3 0
```
output
```
4
```

input
```
5 3
1 5 10 14 17
4 11 15
```
output
```
3
```

D. Road to Post Office

1 second, 256 megabytes

Vasiliy has a car and he wants to get from home to the post office. The distance which he needs to pass equals to d kilometers.

Vasiliy's car is not new — it breaks after driven every k kilometers and Vasiliy needs t seconds to repair it. After repairing his car Vasiliy can drive again (but after k kilometers it will break again, and so on). In the beginning of the trip the car is just from repair station.

To drive one kilometer on car Vasiliy spends a seconds, to walk one kilometer on foot he needs b seconds ($a < b$).

Your task is to find minimal time after which Vasiliy will be able to reach the post office. Consider that in every moment of time Vasiliy can left his car and start to go on foot.

Input

The first line contains 5 positive integers d, k, a, b, t ($1 \le d \le 10^{12}$; $1 \le k, a, b, t \le 10^6$; $a < b$), where:

- d — the distance from home to the post office;
- k — the distance, which car is able to drive before breaking;
- a — the time, which Vasiliy spends to drive 1 kilometer on his car;
- b — the time, which Vasiliy spends to walk 1 kilometer on foot;
- t — the time, which Vasiliy spends to repair his car.

Output

Print the minimal time after which Vasiliy will be able to reach the post office.

input
```
5 2 1 4 10
```
output
```
14
```

input
5 2 1 4 5
output
13

In the first example Vasiliy needs to drive the first 2 kilometers on the car (in 2 seconds) and then to walk on foot 3 kilometers (in 12 seconds). So the answer equals to 14 seconds.

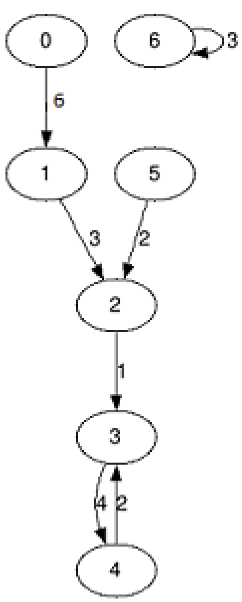
In the second example Vasiliy needs to drive the first 2 kilometers on the car (in 2 seconds), then repair his car (in 5 seconds) and drive 2 kilometers more on the car (in 2 seconds). After that he needs to walk on foot 1 kilometer (in 4 seconds). So the answer equals to 13 seconds.

E. Analysis of Pathes in Functional Graph

2 seconds, 512 megabytes

You are given a *functional graph*. It is a directed graph, in which from each vertex goes exactly one arc. The vertices are numerated from 0 to $n - 1$.

Graph is given as the array $f_0, f_1, ..., f_{n-1}$, where f_i — the number of vertex to which goes the only arc from the vertex i. Besides you are given array with weights of the arcs $w_0, w_1, ..., w_{n-1}$, where w_i — the arc weight from i to f_i.



The graph from the first sample test.

Also you are given the integer k (the length of the path) and you need to find for each vertex two numbers s_i and m_i, where:

- s_i — the sum of the weights of all arcs of the path with length equals to k which starts from the vertex i;
- m_i — the minimal weight from all arcs on the path with length k which starts from the vertex i.

The length of the path is the number of arcs on this path.

Input

The first line contains two integers n, k ($1 \le n \le 10^5$, $1 \le k \le 10^{10}$). The second line contains the sequence $f_0, f_1, ..., f_{n-1}$ ($0 \le f_i < n$) and the third — the sequence $w_0, w_1, ..., w_{n-1}$ ($0 \le w_i \le 10^8$).

Output

Print n lines, the pair of integers s_i, m_i in each line.

input
7 3 1 2 3 4 3 2 6 6 3 1 4 2 2 3
output
10 1 8 1 7 1 10 2 8 2 7 1 9 3

input
4 4 0 1 2 3 0 1 2 3
output
0 0 4 1 8 2 12 3

input

```
5 3
1 2 3 4 0
4 1 2 14 3
```

output

```
7 1
17 1
19 2
21 3
8 1
```

F. T-Shirts

4 seconds, 1024 megabytes

The big consignment of t-shirts goes on sale in the shop before the beginning of the spring. In all n types of t-shirts go on sale. The t-shirt of the i-th type has two integer parameters — c_i and q_i, where c_i — is the price of the i-th type t-shirt, q_i — is the quality of the i-th type t-shirt. It should be assumed that the unlimited number of t-shirts of each type goes on sale in the shop, but in general the quality is not concerned with the price.

As predicted, k customers will come to the shop within the next month, the j-th customer will get ready to spend up to b_j on buying t-shirts.

All customers have the same strategy. First of all, the customer wants to buy the maximum possible number of the highest quality t-shirts, then to buy the maximum possible number of the highest quality t-shirts from residuary t-shirts and so on. At the same time among several same quality t-shirts the customer will buy one that is cheaper. The customers don't like the same t-shirts, so each customer will not buy more than one t-shirt of one type.

Determine the number of t-shirts which each customer will buy, if they use the described strategy. All customers act independently from each other, and the purchase of one does not affect the purchase of another.

Input

The first line contains the positive integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of t-shirt types.

Each of the following n lines contains two integers c_i and q_i $(1 \le c_i, q_i \le 10^9)$ — the price and the quality of the i-th type t-shirt.

The next line contains the positive integer k $(1 \le k \le 2 \cdot 10^5)$ — the number of the customers.

The next line contains k positive integers $b_1, b_2, ..., b_k$ $(1 \le b_j \le 10^9)$, where the j-th number is equal to the sum, which the j-th customer gets ready to spend on t-shirts.

Output

The first line of the input data should contain the sequence of k integers, where the i-th number should be equal to the number of t-shirts, which the i-th customer will buy.

input

```
3
7 5
3 5
4 3
2
13 14
```

output

```
2 3
```

input

```
2
100 500
50 499
4
50 200 150 100
```

output

```
1 2 2 1
```

In the first example the first customer will buy the t-shirt of the second type, then the t-shirt of the first type. He will spend 10 and will not be able to buy the t-shirt of the third type because it costs 4, and the customer will owe only 3. The second customer will buy all three t-shirts (at first, the t-shirt of the second type, then the t-shirt of the first type, and then the t-shirt of the third type). He will spend all money on it.