DATA STRUCTURES AND ALGORITHMS

ASSIGNMENT-4
Language allowed: C

January 30, 2020

A. All Might vs The League of Villains

The league of villains was finally able to corner the great hero All Might. Underestimating his valor, they all decide to charge at him at once. As All Might's sidekick, you (Dave) see the locations of all the villains and All Might as points (with X and Y coordinates) on your radar. You know that the *bottom-most* point (the minimum y-coordinate) on the radar corresponds to that of All Might. You recommend to fend off the villains in a *in a single counter clockwise sweep* (see sample case for clarity). Help All Might defend himself against the league of villains in a counter clockwise manner. If more than one villain has the same *orientation* with respect to All Might, you suggest All Might to first fend of the villain who is closer to him (in terms of manhattan distance). Assume that All Might does not move from his point.
Note that, for this problem, the direct use of the inbuilt *qsort* function cannot be made.

Input

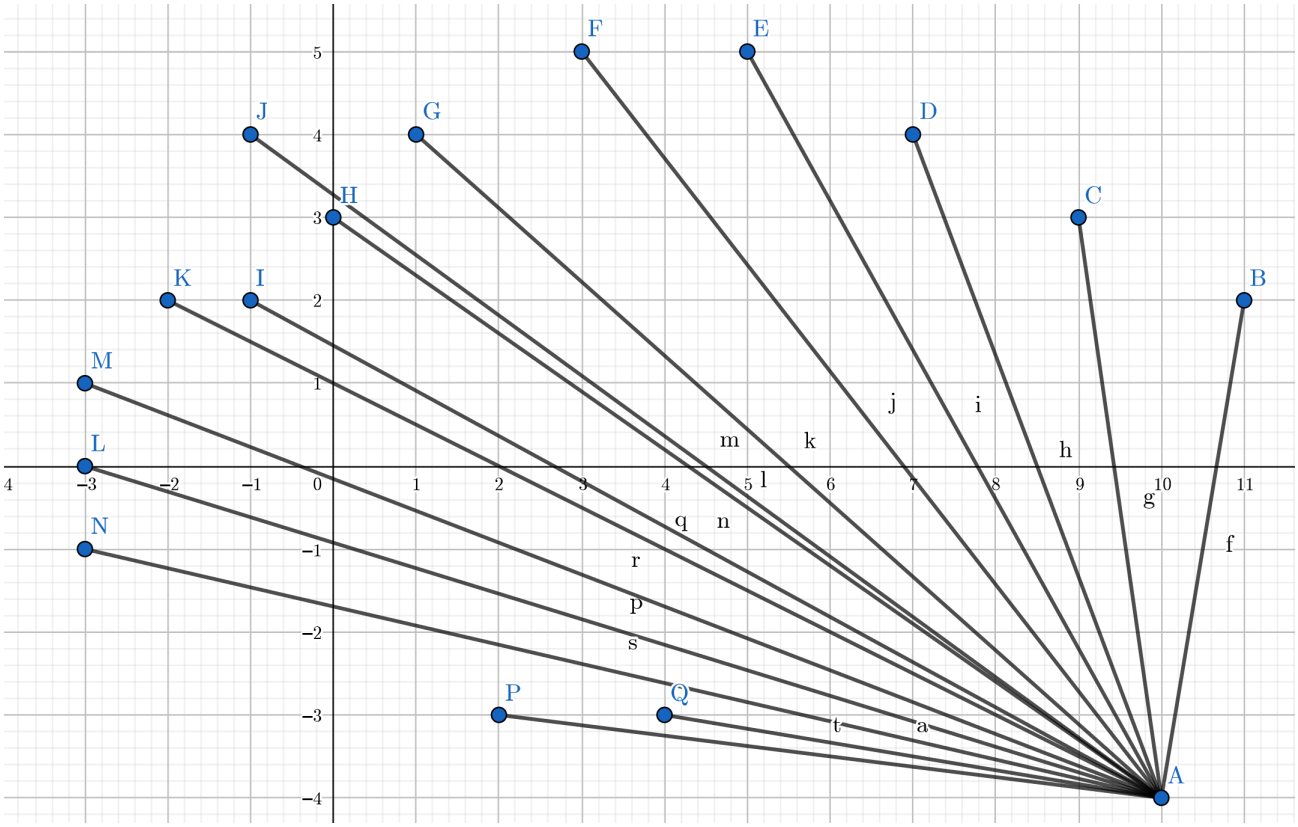
The first line of input contains a single integer N ($2 \leq N \leq 10^5$) denoting the number of points on your radar. Each of the following N lines contain three space-separated integers P, X_P, Y_P ($1 \leq P \leq N$, $-10^9 \leq X_P, Y_P \leq 10^9$) denoting an index number and its corresponding coordinates as you see on your radar. It is guaranteed the coordinates given will result in a unique bottom-most point.

Output

Print a sequence of N-1 integers (B), where B_i denotes the index of the villain he fends off in the i^{th} instant. (in counter-clockwise direction). Figure for sample testcase-1 is given on next page.

```
input
16
12 -3 -1
15 5 5
13 2 -3
14 3 5
1 11 2
2 -1 4
10 -2 2
5 7 4
4 9 3
7 -3 1
6 -3 0
8 10 -4
9 -1 2
16 1 4
3 4 -3
11 0 3

output
1 4 5 15 14 16 2 11 9 10 7 6 12 3 13
```


F
E
J
G
D
H
C
K
I
B
M
j
i
m
k
L
h
l
g
q
n
f
N
r
p
s
P
Q
t
a
A

B. Kalos League

Ash and Greninja are preparing for the Kalos league and but somehow their training is falling short. In order to achieve the *Ash-Greninja* form, the duo need E units of experience. By battling various trainers on their way, they can gain experience points. As Ash believes in the policy of *the more the merrier*, help the duo find the right trainers to battle so that they can gain the necessary E experience points by battling the maximum number of trainers. The duo plan to *immediately* stop battling once they get atleast E experience points and wish to battle a trainer exactly once. If there exist multiple subsets of trainers that let them do so, pick the subset of trainers that results in the smallest combined score.
Note that, for this problem, the direct use of the inbuilt *qsort* function cannot be made.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^5$) denoting the number of trainers around Ash. The next line contains N space-separated integers ($0 \leq A_i \leq 10^9$) with the i^{th} integer (A_i) denoting the experience points the duo will gain by battling the i^{th} trainer. The last line contains a single integer E ($1 \leq \text{E} \leq 10^9$), denoting the experience Ash and Greninja wish to gain.

Output

Output a sequence of integers, denoting the indices (1-indexed) of the trainers Ash and Greninja should battle so that they can gain the necessary experience points, in ascending order of their indices. If more than one trainer gives them the same experience points, they tend to battle the trainer with lower index first. If they can't gain the necessary experience points despite all battles, print "NO CHANCE".

```
input
9
5 5 10 1 2 3 5 9 12
19

output
4 5 6 1 2 7
```

```
input
5
1 1 1 1 1
8

output
NO CHANCE
```

C. Tricky Question

Ameya and Atharv have a tricky question for you. As you have already studied sorting integers, they ask you to write a similar program to sort bigger integers of higher magnitudes. Can you take up the challenge?
Note that, for this problem, the direct use of the inbuilt *qsort* function cannot be made.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^5$) denoting the number of integers you have to sort. Each of the following N lines contain a single integer X_i ($-10^{50} \leq X_i \leq 10^{50}$).

Output

Output the sequence of integers in descending order (one in each line).

```
input
6
93459382187591
124455729347
93123412187591
12988
-1298848528581
-87

output
93459382187591
93123412187591
124455729347
12988
-87
-1298848528581
```

D. Tokyo Tour

You just landed in Tokyo and are quite intrigued by the bus system and the city structure. You realize that all the places (places are numbered from 1 serially) in Tokyo are located on a straight line and adjacent places are a *unit distance* apart. Buses start from place 1 (where you currently are) and stop at particular places (place 1 is obviously one of the stops) only and the bus travel cost is completely free! You plan to visit various parts in Tokyo and decide to take the bus. After you de-bus, you take a cab to the place you want to reach (if the place you want to reach is not the same place where you alight). The cab costs in Tokyo are linear, i.e it is equal to the distance between the initial and final places. As you want to explore Tokyo as much as possible and want to spend *as less money as possible* on cab travels to minimize your expenditure, you decide to write a program that will help you find the cost of your travel.
Note that, for this problem, the direct use of the inbuilt *qsort* function cannot be made.

Input

The first line of input contains two space-separated integers N ($2 \leq N \leq 10^9$) denoting the number of places in Tokyo and M ($1 \leq M \leq min(N,\ 10^5)$) denoting the number of places the bus stops in Tokyo. The next line contains M space-separated integers ($1 \leq A_i \leq N$), denoting the places where the bus stops. The next line contains a single integer Q ($1 \leq Q \leq 10^4$) denoting the number of independent queries for the problem. Each of the following Q lines contain a single integer X_i ($1 \leq X_i \leq N$) denoting the place you want to visit. The places are numbered serially from 1 to N.

Output

Print Q integers (one in each line) where the i^{th} output is the answer to i^{th} query, i.e, the minimum cost you would incur for your travel from place 1 to place X_i.

```
input
20 6
12 8 7 1 3 15
4
8
19
6
10

output
0
4
1
2
```

E. Find those pairs!

In this task, you are supposed to find the number of *inversion pairs* in an array. Two elements of an array A form an inversion pair if they satisfy the relation that $A_i > A_j$ and i < j.
Note that, for this problem, the direct use of the inbuilt *qsort* function cannot be made.

Input

The first line contains a single integer N ($1 \leq N \leq 2.10^5$) denoting the size of the array. The following line contains N space-separated integers denoting the array A ($-10^9 \leq A_i \leq 10^9$).

Output

Print a single integer X, denoting the number of inversion pairs in the given array A.

```
input
5
-3 23 10 7 8

output
5
```

```
input
13
-20 5 1 9 10 19 -45 -12 -13 23 7 8 9

output
29
```

F. Simple Problem

Kirisu-sensei has a very simple problem for Fumiya Furuhashi to boost her confidence. She gives her an array and asks to segregate the elements in such a way that all elements less than a given element K are to its left and all the greater ones are to its right. The elements equal to the chosen element (K) can be either on the left or right. As Furuhashi has recently learnt programming she desires to write a program to solve the above problem. As she is still unsure about her code, she turns to you, Yuiga, for help.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^6$) denoting size of the array. The next line contains N space-separated integers denoting the array A ($-10^9 \leq A_i \leq 10^9$). The next line contains single integer K (K $\in$ A) denoting the element with respect to which the rest of the elements have to segregated.

Output

Print N integers, denoting the array after all segregation are done. If there are multiple answers, print any. If no segregation are required, print "ALL SET".

---

input
```
7
0 5 9 3 4 5 8
3
```

output
```
0 3 5 4 8 9 5
```

---

input
```
10
-1 2 -3 4 5 -6 7 8 9 10
8
```

output
```
ALL SET
```

G. Sort it!

In this task, given a singly linked list, sort it in ascending order in *O(nlogn)* time. Note that no additional arrays or linked lists can be used, i.e, no new linked list/array to hold the answer or manipulate the elements should be made. In addition, simply exchanging the data in the nodes of the linked list is not allowed. Pointer manipulation must be performed to sort the list.

Input

The only line of input contains a sequence of integers A ($-10^9 \leq A_i \leq 10^9$) denoting the list that needs to be sorted. It is guaranteed that the size of the list will be less than 10^4. EOF can be used to mark the end of input.

Output

Print a sequence of integers denoting the list after sorting it.

```
input
-3 -5 0 6 8 12 87 12

output
-5 -3 0 6 8 12 12 87
```

H. Halley and her Piggy-banks

Raj gifted his goddaughter Halley a set of N empty piggy-banks for her first birthday. Since then on, Halley decided to save her pocket money (generously given by Howard, Bernedette and Raj) in them. Everyday, she deposits X dollars into every piggy-bank belonging to some non-empty contiguous subset of the N piggy-banks. After several days, Halley wishes to know the money she has accumulated in each piggybank. Howard, due to his carelessness, has misplaced the record book in which he noted the daily deposits. But luckily, you (Sheldon) remember all the details with your eidetic memory. Help Halley know how much she has accumulated in each of her piggy-banks.

Input

The first line contains two space-separated integers N ($2 \leq N \leq 10^6$) denoting the number of piggy-banks Raj gifted Halley and M ($1 \leq M \leq 10^4$) denoting the number of days Halley saved her money in those piggy-banks. Each of the following M lines contain three space-separated integers L_i, R_i, X_i ($0 \leq L_i < R_i \leq$ N-1, $0 \leq X_i \leq 10^5$) denoting that Halley saved X_i dollars in each of the piggy-bank belonging to the contiguous subset of piggy-banks numbered from L_i to R_i.

Output

Print N integers, denoting the money that Halley saved in total in each piggy bank (the i^{th} integer should denote the total money saved in the i^{th} piggy-bank).

```
input
5 3
0 2 5
1 3 4
0 4 1

output
6 10 10 5 1
```

I. Meeting Room

You are given a meeting room which can only hold *one meeting at a time*. However, you are getting a lot of requests to book the meeting room. You need to find the *maximum* number of meetings that you can hold in the meeting room given the starting and ending times of each of meetings, such that there is no overlap (in the times) between any two meetings. Note that, if a meeting ends at a time then you cannot start another meeting at the same time. Assume that all the bookings are requested at once.

Input

The first line contains a single integer N ($1 \leq N \leq 10^5$) denoting the number of meetings. Each of the following N lines contain two space-separated integers S_i and E_i ($1 \leq S_i < E_i \leq 10^9$) denoting the starting and ending times (the i^{th} line has the timings of the i^{th} meeting).

Output

Print a single integer X, denoting the *maximum* number of meetings you can hold in the meeting room without any overlap.

```
input
5
1 3
4 7
2 4
6 10
8 10

output
3
```

J. Square Root

In this question, you have to find the square-root of a given number. For the integers that are not perfect squares, you are expected to find the *integral part* of the square-root. You are not supposed to use the inbuilt *sqrt* method to do the same.

Input

The first line of input contains a single integer Q ($1 \leq Q \leq 1000$) denoting the number of queries for the problem. Each of the following Q lines contain a single integer N_i ($0 \leq N_i \leq 10^{18}$) denoting the number for which the square-root has to be found.

Output

Print Q integers (one per line), where the i^{th} output denotes the answer (square-root) of the i^{th} query.

```
input
5
4
361
56
2565
2025

output
2
19
7
50
45
```
