Problem A. Binary Search

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Binary search is one of the **fundamental** algorithms in computer science. Binary Search is used to find a value in a **sorted** sequence, in its simplest form, binary search is used to **quickly** find a value in a sorted sequence (consider a sequence an ordinary array). The **sorted sequence** is called the **Search Space** . At each step, the algorithm compares the **median** value in the search space to the **target** value. Based on the comparison and because the sequence is sorted, it can then **eliminate half** of the search space. By doing this **repeatedly** , it will eventually be left with a search space consisting of a single element, the **target** value.

Given n $(1 \leq n \leq 2.10^5)$ which represent the number of elements in the sequence (or array), the elements of the array a $(-10^9 \leq a_i \leq 10^9)$, you are also given q queries $(1 \leq q \leq 10^5)$ for each query find how many numbers in the array are greater than the given number m $(-10^9 \leq m \leq 10^9)$.

The Naive approach is to use linear search which in this case will yield TLE (Time Limit Exceeded).

In order to solve the question, **first** you need to sort the array and this can be done using your own implementation of sort or you can use the built-in standard library in c++ **sort()** .

secondly , we execute the binary search to find how many elements are greater than the given query. There are two methods for implementing Binary Search: your own implementation or using standard library functions. You are always better off using standard library function whenever possible.

```
binary_search(A, target):
   lo = 1, hi = size(A)
   while lo <= hi:
      mid = lo + (hi-lo)/2
      if A[mid] == target:
         return mid
      else if A[mid] < target:
         lo = mid+1
      else:
         hi = mid-1

   // target was not found
```

Using standard library is recommended whenever possible since implementing binary search on your own can be tricky.

In c++, **lower_bound() and upper_bound()** are binary search functions. lower_bound and upper_bound return an iterator or pointer therefore we substract iterator from inputArray to get the index.

$$indexOfNumberGreaterthanQuery = upper\_bound(inputArray, inputArray{+}n, queryValue){-}inputArray$$

$$numberOfElementsGreaterThanQuery = n - index$$

C++ Implementation

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int n, a[2*100000], q, queryValue;
    int index;
    cin >> n ;
    for(int i=0;i<n;i++){
        cin >> a[i];
    }
    sort(a,a+n);
    cin >> q ;
    for(int i=0;i<q;i++){
        cin >> queryValue;
        index = upper_bound(a,a+n,queryValue) - a;
        cout << n - index << endl;
    }
    return 0;
}
```

Java Implementation

```
import java.io.PrintStream;
import java.util.Arrays;
import java.util.Scanner;
import java.util.TreeSet;

public class Main{
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        PrintStream out = System.out;
        int n = in.nextInt();
        int[] inputArrays = new int[n];
        for(int i=0; i<n ; i++){
            inputArrays[i] = in.nextInt();
        }
        Arrays.sort(inputArrays);
        int q = in.nextInt();
        while(q>0){
            int query = in.nextInt();
            int index = higherThanQuery(0,n-1,query,inputArrays);
            out.println(n - index);
            q--;
        }
    }
    public static int higherThanQuery(int first, int last, int target,int array[]){
        int tmp ;
        int counter, step ;
        counter = last+1 - first;
        while(counter>0){
            tmp = first ;
            step = counter/2;
            tmp = tmp + step ;
            if ( !(target < array[tmp]) ){
                first = ++tmp;
                counter -= step + 1;
            }
            else counter = step ;
        }
        return first;
    }
}
```

Input

The first line contains an integer n ($1 \leq n \leq 2.10^5$) representing the number of array elements. The next line contains n integers a_i ($-10^9 \leq a_i \leq 10^9$). The third line contains an integer q representing the number of queries. The following q lines contains single integer representing the queried number m ($-10^9 \leq m \leq 10^9$).

Output

The output consist of q lines. On each line, output the number of integers greater than the given query.

Example

standard input	standard output
8	2
20 20 20 30 30 10 10 10	2
10	2
20	2
21	2
22	2
23	2
24	2
25	2
26	5
27	
28	
10	

Note

For the given example, you are given 8 numbers.

The numbers are: 20 20 20 30 30 10 10 10

and you are given 10 queries.

The first thing is to sort the numbers in ascending order.

10 10 10 20 20 20 30 30

For each query:

20 —-> How many numbers greater than 20, the answer is 2 numbers which are (30,30)

21 —-> How many numbers greater than 21, the answer is 2 numbers which are (30,30)

22 —-> How many numbers greater than 22, the answer is 2 numbers which are (30,30)

23 —-> How many numbers greater than 23, the answer is 2 numbers which are (30,30)

24 —-> How many numbers greater than 24, the answer is 2 numbers which are (30,30)

25 —-> How many numbers greater than 25, the answer is 2 numbers which are (30,30)

26 —-> How many numbers greater than 26, the answer is 2 numbers which are (30,30)

27 —-> How many numbers greater than 27, the answer is 2 numbers which are (30,30)

28 —-> How many numbers greater than 28, the answer is 2 numbers which are (30,30)

10 —-> How many numbers greater than 10, the answer is 5 numbers which are (20,20,20,30,30)

Credit to TopCoder Tutorial on Binary Search `https://www.topcoder.com/community/data-science/data-scien` by lovro. Edited by Code Knights Team

Problem B. Zrog the Judge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Representative of Zrog's country didn't do too well in this Olympics, the Olympics Committee of the respective country was very unhappy about it. They decided to do a very thorough talent search in the country for the next Olympics and train them further, every willing participant had to submit their height, weight and the length they can go in one jump. As all these parameters included numbers and only Zrog was known to be good with numbers, they gave him the responsibility to order the list of participants based on these numbers.

Zrog decided, height takes precedence over weight and weight takes precedence over the length of one jump. Also, more height, less weight and longer jump distances are the priority factor. In case all of these are identical for two participants, they will be chosen in the order entered in the list, first come first serve.

Now, as the number of participants is very large, Zrog wants you to help him by writing a program which will order the list according to the above rules.

Input

The first input is an integer n ($1 \le n \le 100000$).

The following n lines each contains three doubles h ($0 < h \le 100$), w ($0 < w \le 100$) and l ($0 < l \le 100$) denoting the height, weight and jump length respectively.

Output

Print out all the n lines in the correct order according to the rules, the numbers should be until 2 decimal points. Check sample output for correct output format.

Example

`standard input`	`standard output`

`4`
`4.20 4.20 2.10`
`9.11 11.90 1.19`
`4.20 4.20 2.00`
`9.11 91.10 0.19`
	`ID:2 H:9.11 W:11.90 L:1.19`
	`ID:4 H:9.11 W:91.10 L:0.19`
	`ID:1 H:4.20 W:4.20 L:2.10`
	`ID:3 H:4.20 W:4.20 L:2.00`

Problem C. Switch Data Transfer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Ahmad is a computer science student and he is experimenting with a new hidden method of data transfer. The idea is that he will record a video of an LED which would turn on and off depending on the data. The LED will change state (on or off) every second depending on the current bit it is transferring. For example, if it is transferring the data (in binary) of 0110, the LED would be off on the first and fourth second, and on on the second and third second.

Ahmad does not know how to use Arduino yet. So, he uses a toggle switch to turn the switch on or off. Because the toggle switch can wear out, he want to know how often he would toggle the switch. Given the initial state of the switch, and an integer (in decimal) to transfer through the video, how many time would he toggle the switch to transfer the integer.

Note that the video is always 32 seconds long and the integer is always transferred as an unsigned binary number.

Input

The input consist of two line. The first line is a string of either "ON"or "OFF"which is the initial state of the switch. The second line contain an integer D ($0 \le D \le 10^9$) which is the integer he wants to transfer.

Output

Output a single number, which is the number of time he need to switch the toggle switch.

Examples

standard input	standard output

`OFF`
`0`
	`0`

`ON`
`2`
	`3`

Note

In the first example, the initial state is off and the binary representation of 0 is all bit off, so Ahmad does not need to toggle the switch at all.

In the second example, the initial state is on and the binary representation of 2 with 32 bit is 000000000000000000000000000010. So (starting from the least significant bit) he first need to turn it back off for the first bit, change it back on for the second bit, and off again for the third bit, so he toggle the switch 3 times.

Problem D. Zrog and Grog

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Zrog and Grog are two frog soldiers protecting the frog village of Ribbit. Their task is to patrol the village from the their enemy, the Snakas. Zrog and Grog are given two position to guard. That means at one time, Zrog will be guarding at one position and Grog will be guarding the other position. For security reason, the protocol of Ribbit village require them to periodically change place. That means Zrog will go to Grog's position and Grog will go to Zrog's position.

Frogs does not have phones. Because of that, they cannot communicate in order to determine when to move. Their only means of synchronization is through accurate timing. To figure-out the timing, they need to figure-out the time taken for the switch to complete.

The two frogs move in steps. At any steps both the frogs move at the same time. However, they can stay in their position if needed, for example waiting for the other frog to move. For magical reason, the only direction the frogs can go to in one step is up, down, left and right. The frog can only move a maximum of one unit distance every step. Frogs are physical beings. That means they can bump each other. Or in another word, in a single step, they cannot swap place. They can do so in multiple step, for example, Zrog move to an empty space, Grog move to Zrog's original position, then Zrog move to Grog's position. Frogs are very efficient and will always use the least total amount of step.

Given a grid that represent the map of the Ribbit, determine the minimum number of steps needed for the switch to happen.

Input

The first line consist of two integer n and m $(0 \leq n \times m \leq 300)$.

The following n lines consist a string of m characters, forming a grid that represent the map.

Character '.' represent empty space, character 'X' represent an obstacle that the frogs cannot move to. Character 'A' represent the first position. Character 'B' represent the second position.

Output

Output a single integer, the minimum number of step required for them to move, or -1 if it is impossible for them to swap position.

Examples

`standard input`	`standard output`

`3 4`
`....`
`A..B`
`....`
	`5`

`3 4`
`XXXX`
`B..A`
`XXXX`
	`-1`

`3 5`
`XXXXX`
`A...B`
`XX.XX`
	`6`

Note

In the first example, the frog in position A and B cannot both go straight to each other as they will bump with one another. One of the frog will need to take two additional step which goes up and then back down or goes down and then back up. Total step taken is 5, for example frog in position B goes up, left, left, left and down.

In the second example, the swap is impossible as they will bump with each other.

In the third example, one of the frog can use the space in the middle of the 'passage' to wait for the other frog to pass through.

Problem E. Naughty Sisters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

MalayMail at Feb 27 One published" in 10 Malaysians under 18 is either overweight or obese." Dr Tee E Siong explains badly eating habits and lack of exercise are the main reasons to this issue, which also later leads to obesity in adults. "Studying does not end at school for most children. It is normally the case to have back-to-back tuition classes on most days of the week. The children are then left with no energy and time to play sports," he said. Father of two naughty identical twines got worried and decided to sent his daughter to send her daughters to a swimming training. But as these twins are so naughty and playful he decided to send them in different times. So first girl goes to the pool every 2 days and the second one every 3 days. However dad noticed in the day 6, which both supposed to be in the pool only one of them attended and sign the attendance for the other one as well. So he decided when to find out what is the first day they both have to go to pool so he can go and let the trainer know they are twins so they can't cheat anymore. Your task is to write a program, which can calculate the first day they both have swimming lessons for different schedule they might have.

Input

Every line consists of two positive integers $<= 10^9$, which shows the first and second twins pattern of attending the pool. The program should stop when two zero are entered.

Output

Print a line consists of the first common day in the pool for both sisters.

Example

standard input	standard output
`2 3`	`6`
`8 12`	`24`
`5 15`	`15`
`0 0`	