A. Tangdi Kabab

2 s., 256 MB

Yatharth and Ritvik go to ANC and order a plate of Tangdi Kabab. Since one plate has 3 pieces, they decide to play a game to decide who gets two pieces and who gets one. Yatharth will give Ritvik two lowercase strings a and b, and Ritvik has to come up with a third string c, if it exists, such that for each i, b_i = $min(a_i, c_i)$, where s_i denotes the $i-$th character of string s. Your task is to help Ritvik and impress him well before you join the CCCC.

Input

The first line of input contains the string a.

The second line of input contains the string b.

$(1 \le len(a) = len(b) \le 100)$

Output

If there is no such string c, print `-1`.

Otherwise, print a string c. In case of more than one answers, any of them work.

input

```
ab
aa
```

output

```
ba
```

input

```
iamlordvoldemort
iamlardolldeddle
```

output

```
tommarvoloriddle
```

input

```
ab
ba
```

output

```
-1
```

For the first test, min('a','b') = 'a' and min('b','a') = 'a'.

Another solution for the second case is "`iamlardolldeddle`".

There is no solution for the third case.

B. Madhav surviving Subham

2 s., 256 MB

Madhav is at the top left corner of a grid, consisting of p rows and q columns, in a cell $(1, 1)$.

In one step, he can move into a cell, adjacent by a side to the current one:

- $(x, y) \to (x, y + 1)$;
- $(x, y) \to (x + 1, y)$;
- $(x, y) \to (x, y - 1)$;
- $(x, y) \to (x - 1, y)$.

Madhav can't go outside the grid.

Subham is standing on cell (s_x, s_y) with a lightsaber. If Madhav comes into some cell that has distance less than or equal to d to Subham, he will die. The distance between two cells (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$.

Madhav has to go from $(1, 1)$ to (p, q). Print the least number of steps that Madhav can take to reach the cell (p, q) without getting killed. Print `-1` if it's impossible to reach the cell (p, q).

Subham is neither in the starting cell nor in the ending cell. The starting cell always has a distance greater than d to Subham.

Input

The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of testcases.

The only line of each testcase contains five integers p, q, s_x, s_y, d $(2 \le p, q \le 1000; 1 \le s_x \le p; 1 \le s_y \le q; 0 \le d \le p + q)$ — the size of the grid, the cell in which Subham is standing and the killing distance Subham has.

Subham is neither in the starting cell, nor in the ending cell ($(s_x, s_y) \ne (1, 1)$ and $(s_x, s_y) \ne (p, q)$). The starting cell $(1, 1)$ always has distance greater than d to Subham ($|s_x - 1| + |s_y - 1| > d$).

Output

For each testcase, print a single integer. If it's possible to reach the cell (p, q) from $(1, 1)$ without getting killed or moving outside the grid, then print the least number of steps for Madhav to reach it. Otherwise, print `-1`.

input

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

output

```
3
-1
8
```

C. A Sleepy Night

1.5 seconds, 256 megabytes

You were feeling sleepy. Noticing this, your roommate decided to help you stay awake because it was still 6 PM only. He decided to give you an array a of N numbers (denoted by $a_i, 1 \le i \le N$) from the top of his head and formulated Q queries to ask you.

Each query consists of a number x, and your task is to estimate the product $P = \prod_{i=1}^{N}(a_i - x)$.

If $P > 0$, it can be estimated to 1. If $P < 0$, it can be estimated to -1. Otherwise, P is 0.

Answer your roommate's queries in order to stay awake till dinner.

Input

The first line of input contains two integers representing $N(1 \le N \le 10^5)$ and Q $(1 \le Q \le 10^5)$ respectively.

The second line contains N space separated integers representing the array a $(-10^9 \le a_i \le 10^9)$.

Q lines follow. Each line contains a single integer x representing a query $(-10^9 \le x \le 10^9)$.

Output

Output a single integer for each query, representing the estimate for that query. The estimate can only be -1, 0 or 1.

```
input
12 7
-64 -32 -16 -8 -4 -2 2 4 6 8 16 32
3
6
9
15
-392
0
321
```

```
output
-1
0
1
1
1
1
1
```

D. Fibonacci Strikes Back

2 s., 256 MB

A fibonacci number is a number that is part of the series $1, 1, 2, 3, 5, \ldots$.

Let us define an **extended Fibonacci series** to be any sequence of **non-decreasing** non-negative integers defined by the relation

$$a_{n+1} = a_n + a_{n-1}$$

with any two non-negative integers a_1, a_2 as initial values.

Vishnu was given two numbers that were part of an extended Fibonacci sequence, but he lost one. He remembers that the number n he still has was previously the kth element of the extended Fibonacci sequence, and number l was the immediate successor of n. Now, he wants to find out what the number l he lost could have been. Please help Vishnu find how many different values l could have taken.

Input

The first line contains an integer t ($1 \leq t \leq 2 \cdot 10^5$), the total number of testcases.

Each test case contains a single line with two integers, n and k ($1 \leq n \leq 2 \cdot 10^5, 3 \leq k \leq 10^9$).

It is guaranteed the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output a newline containing a single integer, the number of possible values of l such that the k-th element in the sequence is n.

```
input
2
1 3
1 4
```

```
output
1
0
```

For the first testcase, the only valid sequence is $[0, 1, 1]$.

For the second testcase, it can be shown that no valid solution exists.

E. Beautiful Permutations

1 s., 256 MB

You are given an array and your task is to find the k^{th} beautiful permutation of given array when all the beautiful permutation of the array are arranged in lexicographic order. It is given that every element of the array is distinct.

The rules for finding the beautiful permutation of an array are :

(i) Even-indexed elements can only be switched with even-indexed elements.

(ii) Odd-indexed elements can only be switched with odd-indexed elements.

A permutation of an array is defined as another array that can be created by rearranging the elements of given array.

Input

The first line contains an integer n $(1 \leq n < 10)$. The size of the array.

The second line contains an array a of size n in which all the elements are different and $0 \leq a_i \leq 10^5$.

The third line contains an integer q $(1 \leq q < 3000)$. The number of queries for different values of k.

The next q lines contains an integer k $(1 \leq k < 3000)$. The rank of the required beautiful permutation of an array.

Output

Output q arrays.

The k^{th} lexicographically beautiful permutation for each k.

It is given that k^{th} rank string always exists.

```
input
2
1 2
1
1
```

```
output
1 2
```

```
input
3
3 5 7
2
1
2
```

```
output
3 5 7
7 5 3
```

In first example, [1,2] has only one beautiful permutation which is [1,2].

In second example, [3,5,7] has two possible beautiful permutation. [3,5,7] is lexicographically smaller than [7,5,3]. So, 1^{st} rank is given to [3,5,7] ad second rank is given to [7,5,3]

F. Ameesh Hates CP

1 s., 256 MB

Ameesh, an insomniac, struggles to sleep peacefully at night. The only way he can fall asleep is by disturbing the tranquility of a non-decreasing array. Given a non-decreasing array, Ameesh can perform the following operation:

Combine two neighboring elements in the array using a specific bitwise operation called XOR. This operation replaces the two elements with the result of their XOR, effectively reducing the length of the array by one. Note that this operation can only be performed if the array has at least two elements.

For instance, if the array is $[2, 3, 4, 5]$ and Ameesh selects 3 and 4, the array becomes $[2, 7, 5]$ $(3 \oplus 4 = 7)$.

To achieve a restful sleep, Ameesh needs to apply this operation a certain number of times until the resulting array is no longer non-decreasing. Help Ameesh determine the resulting array using the fewest number of steps possible, as he has been awake for several consecutive days now.

Input

The input consists of two lines.

The first line contains a single integer n $(2 \leq n \leq 10^5)$, representing the length of the initial array.

The second line contains n space-separated non-negative integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$, denoting the elements of the initial array. It is guaranteed that $a_i \le a_{i+1}$ for all $1 \le i < n$.

Output

Print a single integer — the minimum number of steps needed. If there is no solution, print -1.

input

```
4
2 5 6 8
```

output

```
1
```

input

```
2
6 9
```

output

```
-1
```

input

```
5
1 2 4 6 20
```

output

```
2
```

In the 3rd example, we start by selecting 1 and 2, resulting in the array $[3, 4, 6, 20]$. Next, we choose to remove 3 and 4, leading to $[7, 6, 20]$. This new array is no longer non-decreasing since 7 is greater than 6. It can be shown that it is not possible to achieve a non-decreasing array in lesser operations for the given array.