A. Twice

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

Kinich wakes up to the start of a new day. He turns on his phone, checks his mailbox, and finds a mysterious present. He decides to unbox the present.

Kinich unboxes an array a with n integers. Initially, Kinich's score is 0. He will perform the following operation any number of times:

- Select two indices i and j $(1 \leq i < j \leq n)$ such that neither i nor j has been chosen in any previous operation and $a_i = a_j$. Then, add 1 to his score.

Output the maximum score Kinich can achieve after performing the aforementioned operation any number of times.

Input

The first line contains an integer t ($1 \leq t \leq 500$) — the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 20$) — the length of a.

The following line of each test case contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$).

Output

For each test case, output the maximum score achievable on a new line.

Standard Input	Standard Output

5
1
1
2
2 2
2
1 2
4
1 2 3 1
6
1 2 3 1 2 3
	0
	1
	0
	1
	3

Note

In the first and third testcases, Kinich cannot perform any operations.

In the second testcase, Kinich can perform one operation with $i = 1$ and $j = 2$.

In the fourth testcase, Kinich can perform one operation with $i = 1$ and $j = 4$.

B. Intercepted Inputs

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

To help you prepare for your upcoming Codeforces contest, Citlali set a grid problem and is trying to give you a n by m grid through your input stream. Specifically, your input stream should contain the following:

- The first line contains two integers n and m — the dimensions of the grid.
- The following n lines contain m integers each — the values of the grid.

However, someone has intercepted your input stream, shuffled all given integers, and put them all on one line! Now, there are k integers all on one line, and you don't know where each integer originally belongs. Instead of asking Citlali to resend the input, you decide to determine the values of n and m yourself.

Output any possible value of n and m that Citlali could have provided.

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains an integer k ($3 \leq k \leq 2 \cdot 10^5$) — the total number of inputs in your input stream.

The following line of each test case contains k integers $a_1, a_2, \ldots, a_k$ ($1 \leq a_i \leq k$) — the shuffled inputs of your input stream. It is guaranteed that n and m are contained within the k integers.

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

Output

For each test case, output two integers, one possible value of n and m. If multiple possible answers exist, output any.

Standard Input	Standard Output

5
3
1 1 2
11
3 3 4 5 6 7 8 9 9 10 11
8
8 4 8 3 8 2 8 1
6
2 1 4 5 3 3
8
1 2 6 3 8 5 5 3
	1 1
	3 3
	2 3
	4 1
	1 6

Note

In the first test case, the initial input could have been the following:

```
1 1
```

```
2
```

In the second test case, the initial input could have been the following:

```
3 3
4 5 6
7 8 9
9 10 11
```

,

C. Superultra's Favorite Permutation

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Superultra, a little red panda, desperately wants primogems. In his dreams, a voice tells him that he must solve the following task to obtain a lifetime supply of primogems. Help Superultra!

Construct a permutation* p of length n such that $p_i + p_{i+1}$ is composite† over all $1 \leq i \leq n-1$. If it's not possible, output -1.

*A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (2 appears twice in the array), and $[1,3,4]$ is also not a permutation ($n=3$ but there is 4 in the array).

†An integer x is *composite* if it has at least one other divisor besides 1 and x. For example, 4 is composite because 2 is a divisor.

Input

The first line contains t ($1 \leq t \leq 10^4$) — the number of test cases.

Each test case contains an integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of the permutation.

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

Output

For each test case, if it's not possible to construct p, output -1 on a new line. Otherwise, output n integers $p_1, p_2, \ldots, p_n$ on a new line.

Standard Input	Standard Output
2 3 8	-1 1 8 7 3 6 2 4 5

Note

In the first example, it can be shown that all permutation of size 3 contain two adjacent elements whose sum is prime. For example, in the permutation $[2,3,1]$ the sum $2+3=5$ is prime.

In the second example, we can verify that the sample output is correct because $1+8$, $8+7$, $7+3$, $3+6$, $6+2$, $2+4$, and $4+5$ are all composite. There may be other constructions that are correct.

D. Sharky Surfing

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

Mualani loves surfing on her sharky surfboard!

Mualani's surf path can be modeled by a number line. She starts at position 1, and the path ends at position L. When she is at position x with a jump power of k, she can jump to any **integer** position in the interval $[x, x+k]$. Initially, her jump power is 1.

However, her surf path isn't completely smooth. There are n hurdles on her path. Each hurdle is represented by an interval $[l, r]$, meaning she cannot jump to any position in the interval $[l, r]$.

There are also m power-ups at certain positions on the path. Power-up i is located at position x_i and has a value of v_i. When Mualani is at position x_i, she has the option to collect the power-up to increase her jump power by v_i. There may be multiple power-ups at the same position. When she is at a position with some power-ups, she may choose to take or ignore each individual power-up. No power-up is in the interval of any hurdle.

What is the minimum number of power-ups she must collect to reach position L to finish the path? If it is not possible to finish the surf path, output -1.

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains three integers n, m, and L ($1 \le n, m \le 2 \cdot 10^5, 3 \le L \le 10^9$) — the number of hurdles, the number of power-ups, and the position of the end.

The following n lines contain two integers l_i and r_i ($2 \le l_i \le r_i \le L-1$) — the bounds of the interval for the i'th hurdle. It is guaranteed that $r_i + 1 < l_{i+1}$ for all $1 \le i < n$ (i.e. all hurdles are non-overlapping, sorted by increasing positions, and the end point of a previous hurdle is not consecutive with the start point of the next hurdle).

The following m lines contain two integers x_i and v_i ($1 \le x_i, v_i \le L$) — the position and the value for the i'th power-up. There may be multiple power-ups with the same x. It is guaranteed that $x_i \le x_{i+1}$ for all $1 \le i < m$ (i.e. the power-ups are sorted by non-decreasing position) and no power-up is in the interval of any hurdle.

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

Output

For each test case, output the minimum number of power-ups she must collect to reach position L. If it is not possible, output -1.

Standard Input	Standard Output

4
2 5 50
7 14
30 40
	4
	-1
	1
	2

```
2 2
3 1
3 5
18 2
22 32
4 3 50
4 6
15 18
20 26
34 38
1 2
8 2
10 2
1 4 17
10 14
1 6
1 2
1 2
16 9
1 2 10
5 9
2 3
2 2
```

Note

In the first test case, she can collect power-ups 1, 2, 3, and 5 to clear all hurdles.

In the second test case, she cannot jump over the first hurdle.

In the fourth test case, by collecting both power-ups, she can jump over the hurdle.

E. Kachina's Favorite Binary String

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

This is an interactive problem.

Kachina challenges you to guess her favorite binary string[*] s of length n. She defines $f(l, r)$ as the number of subsequences[†] of `01` in $s_l s_{l+1} \ldots s_r$. **Two subsequences are considered different if they are formed by deleting characters from different positions in the original string, even if the resulting subsequences consist of the same characters.**

To determine s, you can ask her some questions. In each question, you can choose two indices l and r ($1 \le l < r \le n$) and ask her for the value of $f(l, r)$.

Determine and output s after asking Kachina no more than n questions. However, it may be the case that s is *impossible to be determined*. In this case, you would need to report `IMPOSSIBLE` instead.

Formally, s is *impossible to be determined* if after asking n questions, there are always multiple possible strings for s, regardless of what questions are asked. **Note that if you report `IMPOSSIBLE` when there exists a sequence of at most n queries that will uniquely determine the binary string, you will get the Wrong Answer verdict.**

[*] A binary string only contains characters `0` and `1`.

[†] A sequence a is a subsequence of a sequence b if a can be obtained from b by the deletion of several (possibly, zero or all) elements. For example, subsequences of `1011101` are `0`, `1`, `11111`, `0111`, but not `000` nor `11100`.

Input

The first line of input contains a single integer t ($1 \le t \le 10^3$) — the number of test cases.

The first line of each test case contains a single integer n ($2 \le n \le 10^4$) — the length of s.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Interaction

To ask a question, output a line in the following format (do not include quotes)

- "**`? l r`**" ($1 \le l < r \le n$)

The jury will return an integer $f(l, r)$.
When you are ready to print the answer, output a single line in the following format

- If s is *impossible to be determined*, output "**`! IMPOSSIBLE`**"
- Otherwise, output "**`! s`**"

After that, proceed to process the next test case or terminate the program if it was the last test case. Printing the answer does not count as a query.

The interactor is **not** adaptive, meaning that the answer is known before the participant asks the queries and doesn't depend on the queries asked by the participant.

If your program makes more than n queries for one test case, your program should immediately terminate to receive the verdict `Wrong Answer`. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

After printing a query do not forget to output the end of line and flush the output. Otherwise, you may get `Idleness limit exceeded` verdict. To do this, use:

- `fflush(stdout) or cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see the documentation for other languages.

Hacks

To make a hack, use the following format.

The first line should contain a single integer t ($1 \leq t \leq 10^3$) – the number of test cases.

The first line of each test case should contain an integer n ($2 \leq n \leq 10^4$) — the length of s.

The following line should contain s, a binary string of length n.

The sum of n over all test cases should not exceed 10^4.

Standard Input	Standard Output
2 5 4 0 1 2 2 0	? 1 5 ? 2 4 ? 4 5 ? 3 5 ! 01001 ? 1 2 ! IMPOSSIBLE

Note

In the first test case:

In the first query, you ask Kachina for the value of $f(1, 5)$, and she responds with 4 in the input stream.

In the second query, you ask Kachina for the value of $f(2, 4)$. Because there are no subsequences of `01` in the string `100`, she responds with 0 in the input stream.

After asking 4 questions, you report `01001` as s, and it is correct.

In the second test case:

In the first query, you ask Kachina for the value of $f(1, 2)$, and she responds with 0 in the input stream. Notice that this is the only distinct question you can ask.

However, notice that the strings 00 and 11 both have an answer of 0, and it is impossible to differentiate between the two. Therefore, we report `IMPOSSIBLE`.

Please note that this example only serves to demonstrate the interaction format. It is not guaranteed the queries provided are optimal or uniquely determine the answer. However, it can be shown there exists a sequence of at most 5 queries that does uniquely determine sample test case 1.

F. Ardent Flames

Input file:	standard input
Output file:	standard output
Time limit:	4 seconds
Memory limit:	256 megabytes

You have obtained the new limited event character Xilonen. You decide to use her in combat.

There are n enemies in a line. The i'th enemy from the left has health h_i and is currently at position x_i. Xilonen has an attack damage of m, and you are ready to defeat the enemies with her.

Xilonen has a powerful "ground stomp" attack. **Before you perform any attacks**, you select an integer p and position Xilonen there (p can be any integer position, including a position with an enemy currently). Afterwards, for each attack, she deals m damage to an enemy at position p (if there are any), $m - 1$ damage to enemies at positions $p - 1$ and $p + 1$, $m - 2$ damage to enemies at positions $p - 2$ and $p + 2$, and so on. Enemies that are at least a distance of m away from Xilonen take no damage from attacks.

Formally, if there is an enemy at position x, she will deal $\max(0, m - |p - x|)$ damage to that enemy each hit. **Note that you may not choose a different p for different attacks.**

Over all possible p, output the minimum number of attacks Xilonen must perform to defeat at least k enemies. If it is impossible to find a p such that eventually at least k enemies will be defeated, output -1 instead. Note that an enemy is considered to be defeated if its health reaches 0 or below.

Input

The first line contains an integer t ($1 \le t \le 10^4$) – the number of test cases.

The first line of each test case contains three integers n, m, and k ($1 \le k \le n \le 10^5$, $1 \le m \le 10^9$).

The following line contains n integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 10^9$).

The last line of each testcase contains n integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le 10^9$, $x_i < x_{i+1}$ for all $1 \le i < n$)

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output an integer on a new line, the minimum number of attacks that must be performed to defeat at least k enemies. If it is impossible to find a p such that eventually at least k enemies will be defeated, output -1 instead.

Standard Input	Standard Output

6
5 5 3
7 7 7 7 7
1 2 3 4 5
9 5 9
2 4 6 8 10 8 6 4 2
1 2 3 4 5 6 7 8 9
2 10 2
1 1
1 20
	2
	2
	-1
	6969697
	15
	1000000000

2 10 1
69696969 420420420
1 20
2 10 2
10 15
1 19
2 2 2
1000000000 1
1 3	

Note

In the first testcase, it is optimal to select $p = 2$. Each attack, the first enemy takes $5 - |2 - 1| = 4$ damage, the second enemy takes 5 damage, the third enemy takes 4 damage, the fourth enemy takes 3 damage, and the fifth enemy takes 2 damage. After 2 attacks, the first three enemies will be defeated. It can be shown that it is impossible to defeat 3 enemies in less than 2 attacks, no matter which p is selected.

In the second testcase, we must kill all 9 enemies. By selecting $p = 5$, all nine enemies will be defeated in 2 attacks.

In the third testcase, we must kill both enemies. However, it can be shown that no p selected will damage both enemies at the same time, so the answer is -1.

In the fourth testcase, selecting $p = 1$ will enable us to defeat the first enemy in 6969697 attacks.

In the fifth testcase, selecting $p = 10$ will make each enemy take 1 damage per attack. Both enemies will be defeated in 15 attacks.

G. Natlan Exploring

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   256 megabytes
```

You are exploring the stunning region of Natlan! This region consists of n cities, and each city is rated with an attractiveness a_i. A directed edge exists from City i to City j if and only if $i < j$ and $\gcd(a_i, a_j) \neq 1$, where $\gcd(x, y)$ denotes the greatest common divisor (GCD) of integers x and y.

Starting from City 1, your task is to determine the total number of distinct paths you can take to reach City n, modulo $998\,244\,353$. Two paths are different if and only if the set of cities visited is different.

Input

The first line contains an integer n ($2 \le n \le 2 \cdot 10^5$) — the number of cities.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($2 \le a_i \le 10^6$) — the attractiveness of each city.

Output

Output the total number of distinct paths you can take to reach City n, modulo $998\,244\,353$.

Standard Input	Standard Output
5 2 6 3 4 6	5
5 4 196 2662 2197 121	2
7 3 6 8 9 11 12 20	7
2 2 3	0

Note

In the first example, the five paths are the following:

- City 1 → City 5
- City 1 → City 2 → City 5
- City 1 → City 2 → City 3 → City 5
- City 1 → City 2 → City 4 → City 5
- City 1 → City 4 → City 5

In the second example, the two paths are the following:

- City 1 → City 3 → City 5
- City 1 → City 2 → City 3 → City 5