Codeforces Round #698 (Div. 2)

A. Nezzar and Colorful Balls

1 second, 512 megabytes

Nezzar has n balls, numbered with integers $1, 2, \ldots, n$. Numbers $a_1, a_2, \ldots, a_n$ are written on them, respectively. Numbers on those balls form a non-decreasing sequence, which means that $a_i \leq a_{i+1}$ for all $1 \leq i < n$.

Nezzar wants to color the balls using the minimum number of colors, such that the following holds.

- For any color, numbers on balls will form a **strictly increasing sequence** if he keeps balls with this chosen color and discards all other balls.

Note that a sequence with the length at most 1 is considered as a strictly increasing sequence.

Please help Nezzar determine the minimum number of colors.

Input

The first line contains a single integer t $(1 \leq t \leq 100)$ — the number of testcases.

The first line of each test case contains a single integer n $(1 \leq n \leq 100)$.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq n)$. It is guaranteed that $a_1 \leq a_2 \leq \ldots \leq a_n$.

Output

For each test case, output the minimum number of colors Nezzar can use.

input

```
5
6
1 1 1 2 3 4
5
1 1 2 2 3
4
2 2 2 2
3
1 2 3
1
1
```

output

```
3
2
4
1
1
```

Let's match each color with some numbers. Then:

In the first test case, one optimal color assignment is $[1, 2, 3, 3, 2, 1]$.

In the second test case, one optimal color assignment is $[1, 2, 1, 2, 1]$.

B. Nezzar and Lucky Number

1 second, 512 megabytes

Nezzar's favorite digit among $1, \ldots, 9$ is d. He calls a **positive** integer lucky if d occurs at least once in its decimal representation.

Given q integers $a_1, a_2, \ldots, a_q$, for each $1 \leq i \leq q$ Nezzar would like to know if a_i can be equal to a sum of several (one or more) lucky numbers.

Input

The first line contains a single integer t $(1 \leq t \leq 9)$ — the number of test cases.

The first line of each test case contains two integers q and d $(1 \leq q \leq 10^4, 1 \leq d \leq 9)$.

The second line of each test case contains q integers $a_1, a_2, \ldots, a_q$ $(1 \leq a_i \leq 10^9)$.

Output

For each integer in each test case, print "YES" in a single line if a_i can be equal to a sum of lucky numbers. Otherwise, print "NO".

You can print letters in any case (upper or lower).

input

```
2
3 7
24 25 27
10 7
51 52 53 54 55 56 57 58 59 60
```

output

```
YES
NO
YES
YES
YES
NO
YES
YES
YES
YES
YES
YES
NO
```

In the first test case, $24 = 17 + 7$, 27 itself is a lucky number, 25 cannot be equal to a sum of lucky numbers.

C. Nezzar and Symmetric Array

2 seconds, 512 megabytes

Long time ago there was a symmetric array $a_1, a_2, \ldots, a_{2n}$ consisting of $2n$ **distinct integers**. Array $a_1, a_2, \ldots, a_{2n}$ is called symmetric if for each integer $1 \leq i \leq 2n$, there exists an integer $1 \leq j \leq 2n$ such that $a_i = -a_j$.

For each integer $1 \leq i \leq 2n$, Nezzar wrote down an integer d_i equal to the sum of absolute differences from a_i to all integers in a, i. e. $d_i = \sum_{j=1}^{2n} |a_i - a_j|$.

Now a million years has passed and Nezzar can barely remember the array d and totally forget a. Nezzar wonders if there exists any symmetric array a consisting of $2n$ distinct integers that generates the array d.

Input

The first line contains a single integer t $(1 \leq t \leq 10^5)$ — the number of test cases.

The first line of each test case contains a single integer n $(1 \leq n \leq 10^5)$.

The second line of each test case contains $2n$ integers $d_1, d_2, \ldots, d_{2n}$ $(0 \leq d_i \leq 10^{12})$.

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

Output

For each test case, print "YES" in a single line if there exists a possible array a. Otherwise, print "NO".

You can print letters in any case (upper or lower).

input

```
6
2
8 12 8 12
2
7 7 9 11
2
7 11 7 11
1
1 1
4
40 56 48 40 80 56 80 48
6
240 154 210 162 174 154 186 240 174 186 162 210
```

output

```
YES
NO
NO
NO
NO
YES
```

In the first test case, $a = [1, -3, -1, 3]$ is one possible symmetric array that generates the array $d = [8, 12, 8, 12]$.

In the second test case, it can be shown that there is no symmetric array consisting of distinct integers that can generate array d.

D. Nezzar and Board

2 seconds, 512 megabytes

n **distinct** integers $x_1, x_2, \ldots, x_n$ are written on the board. Nezzar can perform the following operation multiple times.

- Select two integers x, y (not necessarily distinct) on the board, and write down $2x - y$. Note that you don't remove selected numbers.

Now, Nezzar wonders if it is possible to have his favorite number k on the board after applying above operation multiple times.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The first line of each test case contains two integers n, k ($2 \le n \le 2 \cdot 10^5$, $-10^{18} \le k \le 10^{18}$).

The second line of each test case contains n distinct integers $x_1, x_2, \ldots, x_n$ ($-10^{18} \le x_i \le 10^{18}$).

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

Output

For each test case, print "YES" on a single line if it is possible to have k on the board. Otherwise, print "NO".

You can print each letter in any case (upper or lower).

input

```
6
2 1
1 2
3 0
2 3 7
2 -1
31415926 27182818
2 1000000000000000000
1 1000000000000000000
2 -1000000000000000000
-1000000000000000000 123
6 80
-5 -20 13 -14 -2 -11
```

output

```
YES
YES
NO
YES
YES
NO
```

In the first test case, the number 1 is already on the board.

In the second test case, Nezzar could perform the following operations to write down $k = 0$ on the board:

- Select $x = 3$ and $y = 2$ and write down 4 on the board.
- Select $x = 4$ and $y = 7$ and write down 1 on the board.
- Select $x = 1$ and $y = 2$ and write down 0 on the board.

In the third test case, it is impossible to have the number $k = -1$ on the board.

E. Nezzar and Binary String

2 seconds, 512 megabytes

Nezzar has a binary string s of length n that he wants to share with his best friend, Nanako. Nanako will spend q days inspecting the binary string. At the same time, Nezzar wants to change the string s into string f during these q days, because it looks better.

It is known that Nanako loves consistency so much. On the i-th day, Nanako will inspect a segment of string s from position l_i to position r_i inclusive. If the segment contains both characters '0' and '1', Nanako becomes unhappy and throws away the string.

After this inspection, at the i-th night, Nezzar can secretly change **strictly less** than half of the characters in the segment from l_i to r_i inclusive, otherwise the change will be too obvious.

Now Nezzar wonders, if it is possible to avoid Nanako being unhappy and at the same time have the string become equal to the string f at the end of these q days and nights.

Input

The first line contains a single integer t ($1 \le t \le 2 \cdot 10^5$) — the number of test cases.

The first line of each test case contains two integers n, q ($1 \le n \le 2 \cdot 10^5$, $0 \le q \le 2 \cdot 10^5$).

The second line of each test case contains a binary string s of length n.

The third line of each test case contains a binary string f of length n.

Then q lines follow, i-th of them contains two integers l_i, r_i ($1 \le l_i \le r_i \le n$) — bounds of the segment, that Nanako will inspect on the i-th day.

It is guaranteed that the sum of n for all test cases doesn't exceed $2 \cdot 10^5$, and the sum of q for all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print "YES" on the single line if it is possible to avoid Nanako being unhappy and have the string f at the end of q days and nights. Otherwise, print "NO".

You can print each letter in any case (upper or lower).

input

```
4
5 2
00000
00111
1 5
1 3
2 1
00
01
1 2
10 6
1111111111
0110001110
1 10
5 9
7 10
1 7
3 5
6 10
5 2
10000
11000
2 5
1 3
```

output

```
YES
NO
YES
NO
```

In the first test case, $\underline{00}000 \to 00\underline{01}1 \to 00111$ is one of the possible sequences of string changes.

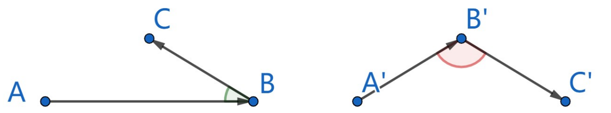
In the second test case, it can be shown that it is impossible to have the string f at the end.

F. Nezzar and Nice Beatmap

2 seconds, 512 megabytes

Nezzar loves the game osu!.

osu! is played on beatmaps, which can be seen as an array consisting of **distinct** points on a plane. A beatmap is called nice if for any three consecutive points A, B, C listed in order, the angle between these three points, centered at B, is **strictly less than** 90 degrees.



Points A, B, C on the left have angle less than 90 degrees, so they can be three consecutive points of a nice beatmap; Points A', B', C' on the right have angle greater or equal to 90 degrees, so they cannot be three consecutive points of a nice beatmap.

Now Nezzar has a beatmap of n **distinct** points $A_1, A_2, \ldots, A_n$. Nezzar would like to reorder these n points so that the resulting beatmap is nice.

Formally, you are required to find a permutation $p_1, p_2, \ldots, p_n$ of integers from 1 to n, such that beatmap $A_{p_1}, A_{p_2}, \ldots, A_{p_n}$ is nice. If it is impossible, you should determine it.

Input

The first line contains a single integer n $(3 \le n \le 5000)$.

Then n lines follow, i-th of them contains two integers x_i, y_i $(-10^9 \le x_i, y_i \le 10^9)$ — coordinates of point A_i.

It is guaranteed that all points are distinct.

Output

If there is no solution, print -1.

Otherwise, print n integers, representing a valid permutation p.

If there are multiple possible answers, you can print any.

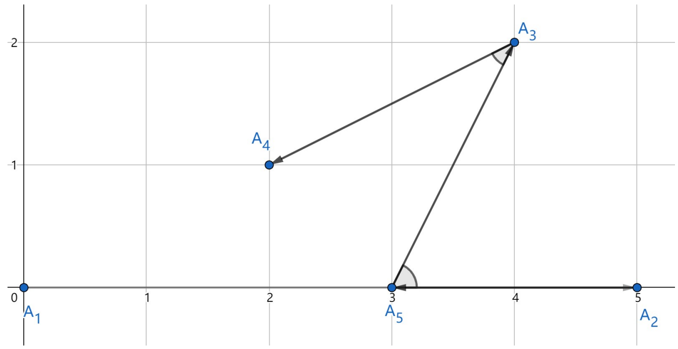
input

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

output

```
1 2 5 3 4
```

Here is the illustration for the first test:



Please note that the angle between A_1, A_2 and A_5, centered at A_2, is treated as 0 degrees. However, angle between A_1, A_5 and A_2, centered at A_5, is treated as 180 degrees.