Codeforces Round #637 (Div. 2) - Thanks, Ivan Belonogov!

A. Nastya and Rice

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Nastya just made a huge mistake and dropped a whole package of rice on the floor. Mom will come soon. If she sees this, then Nastya will be punished.

In total, Nastya dropped n grains. Nastya read that each grain weighs some integer number of grams from $a-b$ to $a+b$, inclusive (numbers a and b are known), and the whole package of n grains weighs from $c-d$ to $c+d$ grams, inclusive (numbers c and d are known). The weight of the package is the sum of the weights of all n grains in it.

Help Nastya understand if this information can be correct. In other words, check whether each grain can have such a mass that the i-th grain weighs some integer number x_i $(a-b \leq x_i \leq a+b)$, and in total they weigh from $c-d$ to $c+d$, inclusive ($c-d \leq \sum\limits_{i=1}^{n} x_i \leq c+d$).

Input

The input consists of multiple test cases. The first line contains a single integer t $(1 \leq t \leq 1000)$ — the number of test cases.

The next t lines contain descriptions of the test cases, each line contains 5 integers: n $(1 \leq n \leq 1000)$ — the number of grains that Nastya counted and a, b, c, d $(0 \leq b < a \leq 1000, 0 \leq d < c \leq 1000)$ — numbers that determine the possible weight of one grain of rice (from $a-b$ to $a+b$) and the possible total weight of the package (from $c-d$ to $c+d$).

Output

For each test case given in the input print "Yes", if the information about the weights is not inconsistent, and print "No" if n grains with masses from $a-b$ to $a+b$ cannot make a package with a total mass from $c-d$ to $c+d$.

Example

input

```
5
7 20 3 101 18
11 11 10 234 2
8 9 7 250 122
19 41 21 321 10
3 10 8 6 1
```

output

```
Yes
No
Yes
No
Yes
```

Note

In the first test case of the example, we can assume that each grain weighs 17 grams, and a pack 119 grams, then really Nastya could collect the whole pack.

In the third test case of the example, we can assume that each grain weighs 16 grams, and a pack 128 grams, then really Nastya could collect the whole pack.

In the fifth test case of the example, we can be assumed that 3 grains of rice weigh 2, 2, and 3 grams, and a pack is 7 grams, then really Nastya could collect the whole pack.

In the second and fourth test cases of the example, we can prove that it is impossible to determine the correct weight of all grains of rice and the weight of the pack so that the weight of the pack is equal to the total weight of all collected grains.

B. Nastya and Door

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

On February 14 Denis decided to give Valentine to Nastya and did not come up with anything better than to draw a huge red heart on the door of the length k ($k \geq 3$). Nastya was very confused by this present, so she decided to break the door, throwing it on the mountains.

Mountains are described by a sequence of heights $a_1, a_2, \dots, a_n$ in order from left to right ($k \le n$). It is guaranteed that neighboring heights are not equal to each other (that is, $a_i \ne a_{i+1}$ for all i from 1 to $n-1$).

Peaks of mountains on the segment $[l, r]$ (from l to r) are called indexes i such that $l < i < r$, $a_{i-1} < a_i$ and $a_i > a_{i+1}$. It is worth noting that the boundary indexes l and r for the segment **are not peaks**. For example, if $n = 8$ and $a = [3, 1, 4, 1, 5, 9, 2, 6]$, then the segment $[1, 8]$ has only two peaks (with indexes 3 and 6), and there are no peaks on the segment $[3, 6]$.

To break the door, Nastya throws it to a segment $[l, l+k-1]$ of consecutive mountains of length k ($1 \le l \le n-k+1$). When the door touches the peaks of the mountains, it breaks into two parts, after that these parts will continue to fall in different halves and also break into pieces when touching the peaks of the mountains, and so on. Formally, the number of parts that the door will break into will be equal to $p+1$, where p is the number of peaks on the segment $[l, l+k-1]$.

Nastya wants to break it into as many pieces as possible. Help her choose such a segment of mountains $[l, l+k-1]$ that the number of peaks on it is maximum. If there are several optimal segments, Nastya wants to find one for which the value l is minimal.

Formally, you need to choose a segment of mountains $[l, l+k-1]$ that has the maximum number of peaks. Among all such segments, you need to find the segment that has the minimum possible value l.

Input

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

The first line of each test case contains two integers n and k ($3 \le k \le n \le 2 \cdot 10^5$) — the number of mountains and the length of the door.

The second line of the input data set contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 10^9$, $a_i \ne a_{i+1}$) — the heights of mountains.

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

Output

For each test case, output two integers t and l — the maximum number of parts that the door can split into, and the left border of the segment of length k that the door should be reset to.

Example

input

```
5
8 6
1 2 4 1 2 4 1 2
5 3
3 2 3 2 1
10 4
4 3 4 3 2 3 2 1 0 1
15 7
3 7 4 8 2 3 4 5 21 2 3 4 2 1 3
7 5
1 2 3 4 5 6 1
```

output

```
3 2
2 2
2 1
3 1
2 3
```

Note

In the first example, you need to select a segment of mountains from 2 to 7. In this segment, the indexes 3 and 6 are peaks, so the answer is 3 (only 2 peaks, so the door will break into 3 parts). It is not difficult to notice that the mountain segments $[1, 6]$ and $[3, 8]$ are not suitable since they only have a 1 peak (for the first segment, the 6 index is not a peak, and for the second segment, the 3 index is not a peak).

In the second example, you need to select a segment of mountains from 2 to 4. In this segment, the index 3 is a peak, so the answer is 2 (only 1 peak, so the door will break into 2 parts).

In the third example, you need to select a segment of mountains from 1 to 4. In this segment, the index 3 is a peak, so the answer is 2 (only 1 peak, so the door will break into 2 parts). You can see that on the segments $[2, 5]$, $[4, 7]$ and $[5, 8]$ the number of peaks is also 1, but these segments have a left border greater than the segment $[1, 4]$, so they are not the correct answer.

C. Nastya and Strange Generator

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

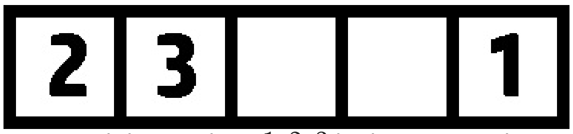
Denis was very sad after Nastya rejected him. So he decided to walk through the gateways to have some fun. And luck smiled at him! When he entered the first courtyard, he met a strange man who was selling something.

Denis bought a mysterious item and it was... Random permutation generator! Denis could not believed his luck.

When he arrived home, he began to study how his generator works and learned the algorithm. The process of generating a permutation consists of n steps. At the i-th step, a place is chosen for the number i $(1 \le i \le n)$. The position for the number i is defined as follows:

- For all j from 1 to n, we calculate r_j — the minimum index such that $j \le r_j \le n$, and the position r_j is not yet occupied in the permutation. If there are no such positions, then we assume that the value of r_j is not defined.
- For all t from 1 to n, we calculate $count_t$ — the number of positions $1 \le j \le n$ such that r_j is defined and $r_j = t$.
- Consider the positions that are still not occupied by permutation and among those we consider the positions for which the value in the $count$ array is maximum.
- The generator selects one of these positions for the number i. The generator can choose **any** position.

Let's have a look at the operation of the algorithm in the following example:



Let $n = 5$ and the algorithm has already arranged the numbers $1, 2, 3$ in the permutation. Consider how the generator will choose a position for the number 4:

- The values of r will be $r = [3, 3, 3, 4, \times]$, where $\times$ means an indefinite value.
- Then the $count$ values will be $count = [0, 0, 3, 1, 0]$.
- There are only two unoccupied positions in the permutation: 3 and 4. The value in the $count$ array for position 3 is 3, for position 4 it is 1.
- The maximum value is reached only for position 3, so the algorithm will uniquely select this position for number 4.

Satisfied with his purchase, Denis went home. For several days without a break, he generated permutations. He believes that he can come up with random permutations no worse than a generator. After that, he wrote out the first permutation that came to mind $p_1, p_2, \ldots, p_n$ and decided to find out if it could be obtained as a result of the generator.

Unfortunately, this task was too difficult for him, and he asked you for help. It is necessary to define whether the written permutation could be obtained using the described algorithm if the generator always selects the position Denis needs.

Input

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

The first line of the test case contains a single integer n $(1 \le n \le 10^5)$ — the size of the permutation.

The second line of the test case contains n different integers $p_1, p_2, \ldots, p_n$ $(1 \le p_i \le n)$ — the permutation written by Denis.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5.

Output

Print "Yes" if this permutation could be obtained as a result of the generator. Otherwise, print "No".

All letters can be displayed in any case.

Example

input

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

output

```
Yes
Yes
No
Yes
No
```

Note

Let's simulate the operation of the generator in the first test.

At the 1 step, $r = [1, 2, 3, 4, 5], count = [1, 1, 1, 1, 1]$. The maximum value is reached in any free position, so the generator can choose a random position from 1 to 5. In our example, it chose 5.

At the 2 step, $r = [1, 2, 3, 4, \times], count = [1, 1, 1, 1, 0]$. The maximum value is reached in positions from 1 to 4, so the generator can choose a random position among them. In our example, it chose 1.

At the 3 step, $r = [2, 2, 3, 4, \times], count = [0, 2, 1, 1, 0]$. The maximum value is 2 and is reached only at the 2 position, so the generator will choose this position.

At the 4 step, $r = [3, 3, 3, 4, \times], count = [0, 0, 3, 1, 0]$. The maximum value is 3 and is reached only at the 3 position, so the generator will choose this position.

At the 5 step, $r = [4, 4, 4, 4, \times], count = [0, 0, 0, 4, 0]$. The maximum value is 4 and is reached only at the 4 position, so the generator will choose this position.

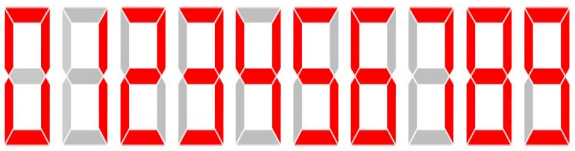
In total, we got a permutation of $2, 3, 4, 5, 1$, that is, a generator could generate it.

D. Nastya and Scoreboard

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Denis, after buying flowers and sweets (you will learn about this story in the next task), went to a date with Nastya to ask her to become a couple. Now, they are sitting in the cafe and finally... Denis asks her to be together, but ... Nastya doesn't give any answer.

The poor boy was very upset because of that. He was so sad that he punched some kind of scoreboard with numbers. The numbers are displayed in the same way as on an electronic clock: each digit position consists of 7 segments, which can be turned on or off to display different numbers. The picture shows how all 10 decimal digits are displayed:



After the punch, some segments stopped working, that is, some segments might stop glowing if they glowed earlier. But Denis remembered how many sticks were glowing and how many are glowing now. Denis broke **exactly** k segments and he knows which sticks are working now. Denis came up with the question: what is the maximum possible number that can appear on the board if you turn on exactly k sticks (which are off now)?

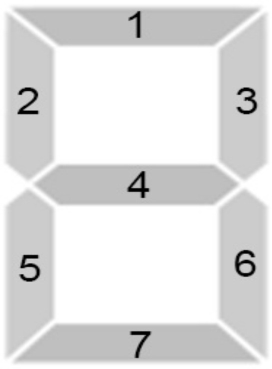
It is **allowed** that the number includes leading zeros.

Input

The first line contains integer n $(1 \leq n \leq 2000)$ — the number of digits on scoreboard and k $(0 \leq k \leq 2000)$ — the number of segments that stopped working.

The next n lines contain one binary string of length 7, the i-th of which encodes the i-th digit of the scoreboard.

Each digit on the scoreboard consists of 7 segments. We number them, as in the picture below, and let the i-th place of the binary string be 0 if the i-th stick is not glowing and 1 if it is glowing. Then a binary string of length 7 will specify which segments are glowing now.



Thus, the sequences "1110111", "0010010", "1011101", "1011011", "0111010", "1101011", "1101111", "1010010", "1111111", "1111011" encode in sequence all digits from 0 to 9 inclusive.

Output

Output a single number consisting of n digits — the maximum number that can be obtained if you turn on exactly k sticks or -1, if it is impossible to turn on exactly k sticks so that a correct number appears on the scoreboard digits.

Examples

input

```
1 7
0000000
```

output

```
8
```

input

```
2 5
0010010
0010010
```

output

```
97
```

input

```
3 5
0100001
1001001
1010011
```

output

```
-1
```

Note

In the first test, we are obliged to include all 7 sticks and get one 8 digit on the scoreboard.

In the second test, we have sticks turned on so that units are formed. For 5 of additionally included sticks, you can get the numbers 07, 18, 34, 43, 70, 79, 81 and 97, of which we choose the maximum -97.

In the third test, it is impossible to turn on exactly 5 sticks so that a sequence of numbers appears on the scoreboard.

E. Nastya and Unexpected Guest

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

If the girl doesn't go to Denis, then Denis will go to the girl. Using this rule, the young man left home, bought flowers and went to Nastya.

On the way from Denis's house to the girl's house is a road of n lines. This road can't be always crossed in one green light. Foreseeing this, the good mayor decided to place safety islands in some parts of the road. Each safety island is located after a line, as well as at the beginning and at the end of the road. Pedestrians can relax on them, gain strength and wait for a green light.

Denis came to the edge of the road exactly at the moment when the green light turned on. The boy knows that the traffic light first lights up g seconds green, and then r seconds red, then again g seconds green and so on.

Formally, the road can be represented as a segment $[0, n]$. Initially, Denis is at point 0. His task is to get to point n in the shortest possible time.

He knows many different integers $d_1, d_2, \ldots, d_m$, where $0 \le d_i \le n$ — are the coordinates of points, in which the safety islands are located. Only at one of these points, the boy can be at a time when the red light is on.

Unfortunately, Denis isn't always able to control himself because of the excitement, so some restrictions are imposed:

- He must always move while the green light is on because it's difficult to stand when so beautiful girl is waiting for you. Denis can change his position by ± 1 in 1 second. While doing so, he must always stay inside the segment $[0, n]$.
- He can change his direction only on the safety islands (because it is safe). This means that if in the previous second the boy changed his position by $+1$ and he walked on a safety island, then he can change his position by ± 1. Otherwise, he can change his position only by $+1$. Similarly, if in the previous second he changed his position by -1, on a safety island he can change position by ± 1, and at any other point by -1.
- At the moment when the red light is on, the boy must be on one of the safety islands. He can continue moving in any direction when the green light is on.

Denis has crossed the road as soon as his coordinate becomes equal to n.

This task was not so simple, because it's possible that it is impossible to cross the road. Since Denis has all thoughts about his love, he couldn't solve this problem and asked us to help him. Find the minimal possible time for which he can cross the road according to these rules, or find that it is impossible to do.

Input

The first line contains two integers n and m $(1 \le n \le 10^6, 2 \le m \le min(n+1, 10^4))$ — road width and the number of safety islands.

The second line contains m distinct integers $d_1, d_2, \ldots, d_m$ $(0 \le d_i \le n)$ — the points where the safety islands are located. It is guaranteed that there are 0 and n among them.

The third line contains two integers g, r $(1 \le g, r \le 1000)$ — the time that the green light stays on and the time that the red light stays on.

Output

Output a single integer — the minimum time for which Denis can cross the road with obeying all the rules.

If it is impossible to cross the road output -1.

Examples

input
```
15 5
0 3 7 14 15
11 11
```
output
```
45
```

input
```
13 4
0 3 7 13
9 9
```
output
```
-1
```

Note

In the first test, the optimal route is:

- for the first green light, go to 7 and return to 3. In this case, we will change the direction of movement at the point 7, which is allowed, since there is a safety island at this point. In the end, we will be at the point of 3, where there is also a safety island. The next 11 seconds we have to wait for the red light.
- for the second green light reaches 14. Wait for the red light again.
- for 1 second go to 15. As a result, Denis is at the end of the road.

In total, 45 seconds are obtained.

In the second test, it is impossible to cross the road according to all the rules.

F. Nastya and Time Machine

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Denis came to Nastya and discovered that she was not happy to see him... There is only one chance that she can become happy. Denis wants to buy all things that Nastya likes so she will certainly agree to talk to him.

The map of the city where they live has a lot of squares, some of which are connected by roads. There is exactly one way between each pair of squares which does not visit any vertex twice. It turns out that the graph of the city is a tree.

Denis is located at vertex 1 at the time 0. He wants to visit every vertex at least once and get back as soon as possible.

Denis can walk one road in 1 time. Unfortunately, the city is so large that it will take a very long time to visit all squares. Therefore, Denis took a desperate step. He pulled out his pocket time machine, which he constructed in his basement. With its help, Denis can change the time to any non-negative time, which is less than the current time.

But the time machine has one feature. If the hero finds himself in the same place and at the same time twice, there will be an explosion of universal proportions and Nastya will stay unhappy. Therefore, Denis asks you to find him a route using a time machine that he will get around all squares and will return to the first and at the same time the maximum time in which he visited any square will be minimal.

Formally, Denis's route can be represented as a sequence of pairs: $\{v_1, t_1\}, \{v_2, t_2\}, \{v_3, t_3\}, \ldots, \{v_k, t_k\}$, where v_i is number of square, and t_i is time in which the boy is now.

The following conditions must be met:

- The route starts on square 1 at time 0, i.e. $v_1 = 1, t_1 = 0$ and ends on the square 1, i.e. $v_k = 1$.
- All transitions are divided into two types:
  1. Being in the square change the time: $\{v_i, t_i\} \to \{v_{i+1}, t_{i+1}\} : v_{i+1} = v_i, 0 \leq t_{i+1} < t_i$.
  2. Walk along one of the roads: $\{v_i, t_i\} \to \{v_{i+1}, t_{i+1}\}$. Herewith, v_i and v_{i+1} are connected by road, and $t_{i+1} = t_i + 1$
- All pairs $\{v_i, t_i\}$ must be different.
- All squares are among $v_1, v_2, \ldots, v_k$.

You need to find a route such that the maximum time in any square will be minimal, that is, the route for which $\max(t_1, t_2, \ldots, t_k)$ will be the minimum possible.

Input

The first line contains a single integer n $(1 \leq n \leq 10^5)$ — the number of squares in the city.

The next $n - 1$ lines contain two integers u and v $(1 \leq v, u \leq n, u \neq v)$ - the numbers of the squares connected by the road.

It is guaranteed that the given graph is a tree.

Output

In the first line output the integer k $(1 \le k \le 10^6)$ — the length of the path of Denis.

In the next k lines output pairs v_i, t_i — pairs that describe Denis's route (as in the statement).

All route requirements described in the statements must be met.

It is guaranteed that under given restrictions there is at least one route and an answer whose length does not exceed 10^6. If there are several possible answers, print any.

Example

input

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

output

```
13
1 0
2 1
3 2
3 1
2 2
4 3
4 1
5 2
5 1
4 2
2 3
2 0
1 1
```