Problem Satan. Draw Polygon Lines

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

This is an interactive problem.

You are given n points $A_i = (x_i, y_i)$ on the plane. It is known that all x_i are distinct and all y_i are distinct.

Your task is to draw polygonal lines connecting these n points.

A polygonal line is defined by a permutation $p_1, p_2, \ldots, p_n$ of numbers from 1 to n. The polygonal line consists of $n-1$ segments, the first segment connects points A_{p_1} and A_{p_2}, the second segment connects points A_{p_2} and A_{p_3}, ..., the last segment connects points $A_{p_{n-1}}$ and A_{p_n}. Note that segments may intersect.

The *sharpness* of a polygonal line is defined as the number of indices $2 \le i \le n-1$ such that the angle $\angle A_{p_{i-1}} A_{p_i} A_{p_{i+1}}$ is acute, i.e., strictly less than $90°$.

You need to solve four tasks:

1. Find any polygonal line that has the maximum possible sharpness.
2. Given an integer c. Find any polygonal line whose sharpness is $\le c$.
3. Given an integer c.

   Answer q queries, each specified by a single integer k_i ($c \le k_i \le n-c$). In the i-th query, you need to construct a polygonal line that has sharpness exactly k_i.
4. Given an integer c.

   For each k from c to $n-c$, construct a polygonal line $p^{(k)}$ with sharpness exactly k. Provide $n-2c+1$ numbers $\text{hash}\left(p^{(c)}\right), \text{hash}\left(p^{(c+1)}\right), \ldots, \text{hash}\left(p^{(n-c)}\right)$ as the answer, where $hash(p) = \left(\sum_{i=1}^{n} p_i b^{i-1}\right) \bmod m$ is the polynomial hash of permutation p with parameters $b = 10^6+3$ and $m = 10^9+7$.

   Then answer q queries, each specified by a single integer k_i ($c \le k_i \le n-c$). In the i-th query, you need to provide the polygonal line $p^{(k_i)}$. It will be checked that the sharpness of this polygonal line is exactly k_i and its hash matches the previously provided value $\text{hash}\left(p^{(k_i)}\right)$.

   Note that queries will appear after receiving the hashes.

Interaction Protocol

The first line contains two integers task, group ($1 \le \text{task} \le 4$, $0 \le \text{group} \le 21$) — the number of the task to be solved in this test and the test group number.

The second line contains a single integer n ($3 \le n \le 80\,000$) — the number of points on the plane.

Each of the next n lines contains two integers x_i, y_i ($|x_i|, |y_i| \le 10^9$) — the coordinates of the points. It is guaranteed that all x_i are distinct and all y_i are distinct.

If task $= 1$, then the input ends here and you should output any permutation with the maximum possible sharpness. The interaction ends here.

If task $\ne 1$, then the next line contains a single integer c ($2 \le c \le \frac{n}{2}$).

If task $= 2$, then the input ends here and you should output any permutation with sharpness $\le c$. The interaction ends here.

If task $= 4$, your solution should output $n-2c+1$ integers $\text{hash}\left(p^{(c)}\right), \text{hash}\left(p^{(c+1)}\right), \ldots, \text{hash}\left(p^{(n-c)}\right)$, where $0 \le \text{hash}\left(p^{(i)}\right) < 10^9+7$. Note that this should not be done if task $= 3$.

Further interaction occurs only if task $= 3$ or task $= 4$.

The next line contains a single integer q $(1 \le q \le 50)$ — the number of queries.

Then q times, in each line, a query k_i $(c \le k_i \le n - c)$ appears. As a response, you should output a permutation on a separate line. The sharpness of this permutation should be exactly k_i. If task $= 4$, the hash of this permutation should match the previously provided hash.

Since this is an interactive problem, after outputting each line, do not forget to output a newline character and flush the output buffer.

Scoring

The tests for this problem consist of twenty-one groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed.

Group	Points	Constraints				Required Groups	Comment
		task	n	c	Additional constraints		
0	0	–	–	–	–	–	Examples.
1	8	1	$n \le 20\,000$	–	$x_i < x_{i+1}$, $y_i < y_{i+1}$	–	
2	6	1	$n \le 10$	–	random points	–	
3	5	1	$n \le 1000$	–	random points	2	
4	5	1	$n \le 20\,000$	–	random points	2 – 3	
5	6	1	$n \le 20\,000$	–	–	1 – 4	
6	17	2	$n = 80\,000$	$c = 800$	–	–	
7	7	3	$n = 80\,000$	$c = 800$	$x_i < x_{i+1}$, $y_i < y_{i+1}$	–	
8	4	3	$n = 50$	$c = 25$	random points	–	
9	4	3	$n = 200$	$c = 80$	random points	–	
10	4	3	$n = 1000$	$c = 300$	random points	–	
11	3	3	$n = 5000$	$c = 600$	random points	–	
12	3	3	$n = 80\,000$	$c = 35\,000$	random points	–	
13	3	3	$n = 80\,000$	$c = 5000$	random points	12	
14	3	3	$n = 80\,000$	$c = 2000$	–	12 – 13	
15	2	3	$n = 80\,000$	$c = 800$	–	7, 12 – 14	
16	6	4	$n = 80\,000$	$c = 800$	$x_i < x_{i+1}$, $y_i < y_{i+1}$	–	
17	3	4	$n = 5000$	$c = 600$	random points	–	
18	3	4	$n = 80\,000$	$c = 35\,000$	random points	–	
19	3	4	$n = 80\,000$	$c = 5000$	random points	18	
20	3	4	$n = 80\,000$	$c = 2000$	–	18 – 19	
21	2	4	$n = 80\,000$	$c = 800$	–	16, 18 – 20	

In the groups where it is indicated that the points are random, all coordinates of all points x_i, y_i are randomly generated with equal probability in the interval $[-10^9, 10^9]$.

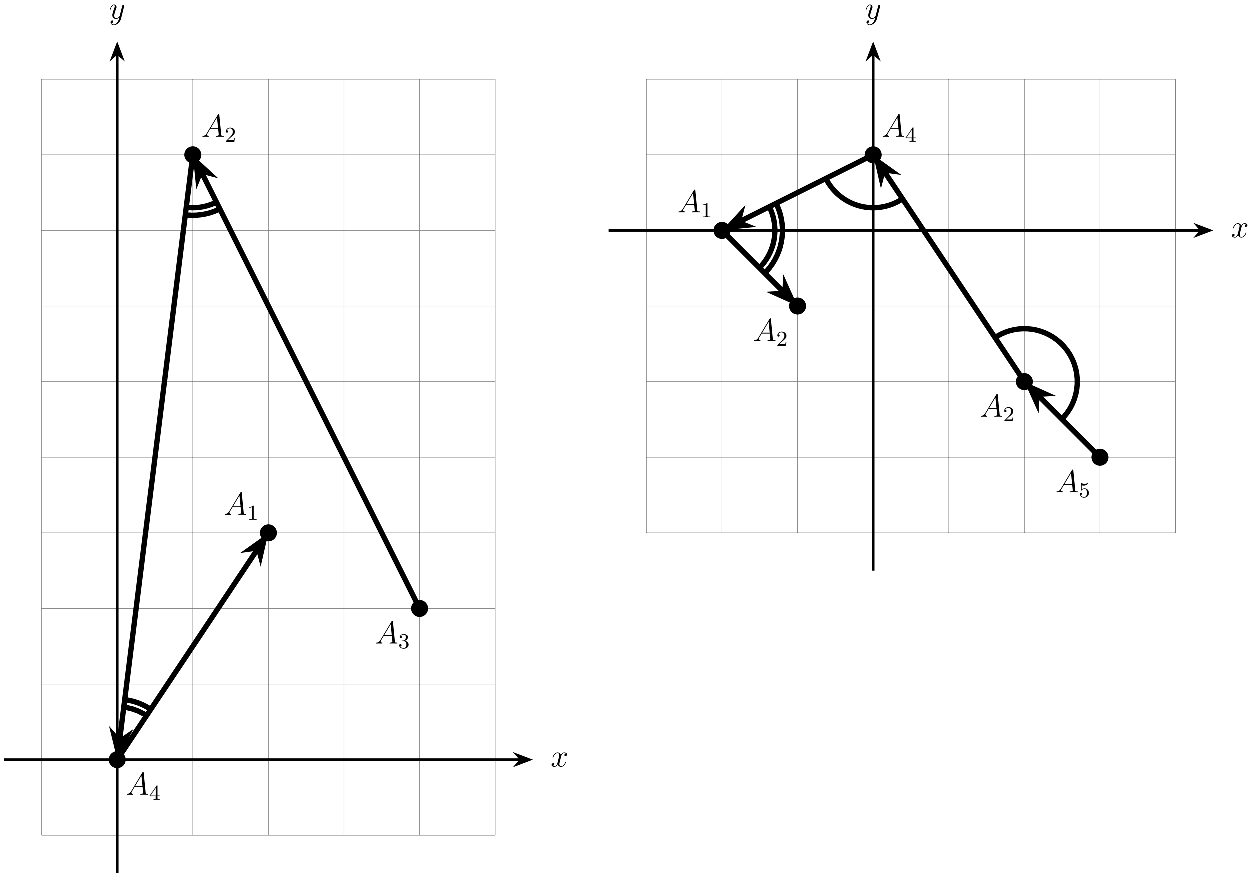
Examples

input	output
1 0 4 2 3 1 8 4 2 0 0	 3 2 4 1
2 0 5 -2 0 -1 -1 0 1 2 -2 3 -3 2	 5 4 3 1 2
3 0 6 0 0 1 1 2 2 3 -3 4 -2 5 -1 2 3 2 3 4	 1 2 3 4 5 6 4 5 6 1 3 2 6 2 4 3 5 1
4 0 5 -2 -1 -1 1 1 6 0 -3 2 0 2 2 2 3	 534735187 776162084 4 5 1 2 3 1 3 2 5 4

Note

In all the figures, acute angles are denoted by two arcs, and non-acute angles are denoted by a single arc.

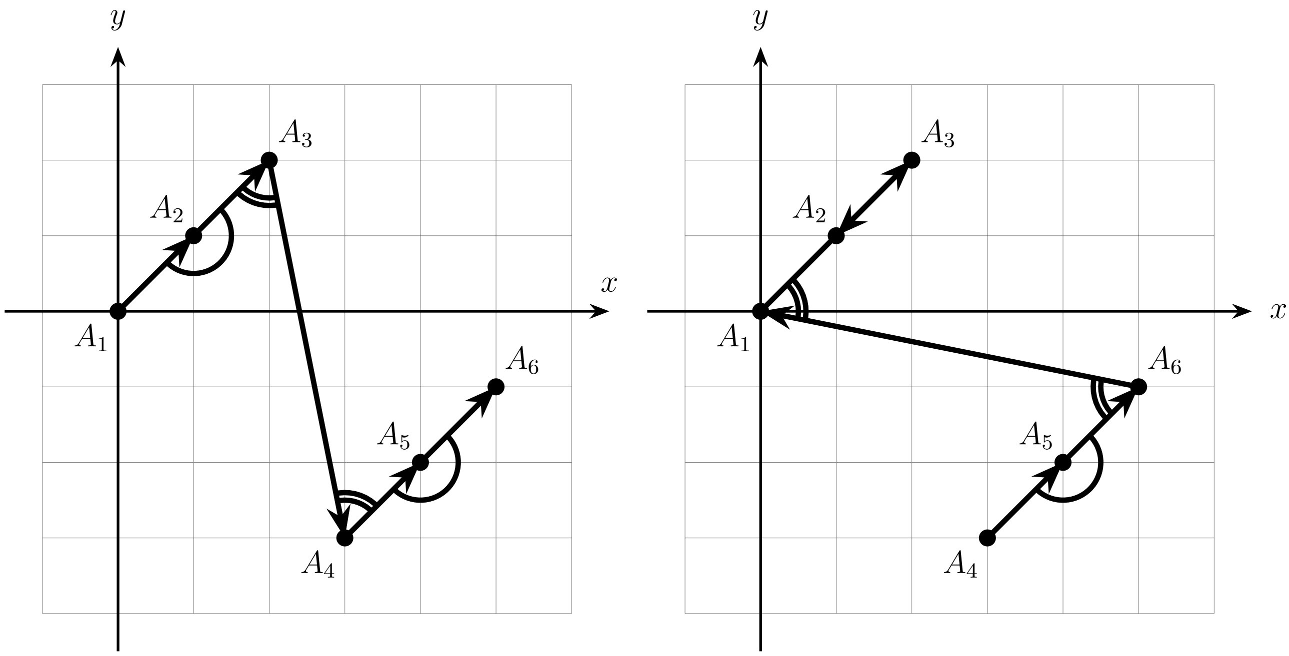
First and second examples

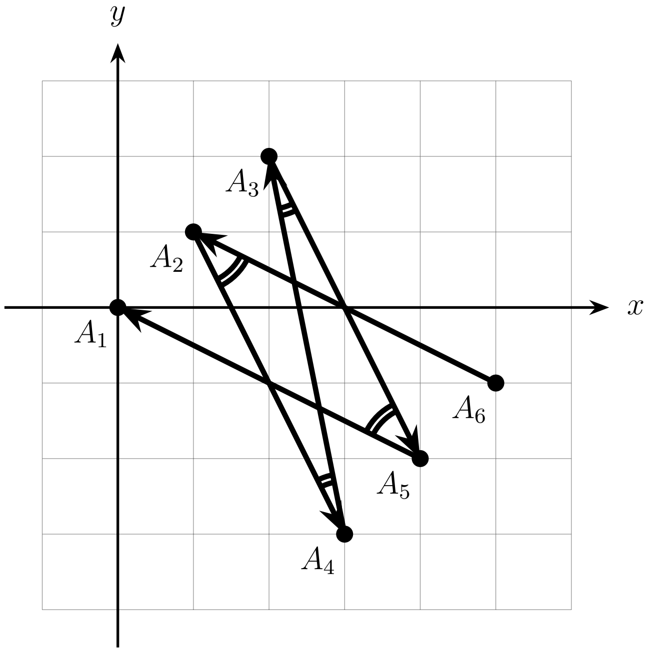


In the first example all angles are sharp, so the line has maximum sharpness 2.

In the second sample the sharpness equals to 1, it is ≤ 2.

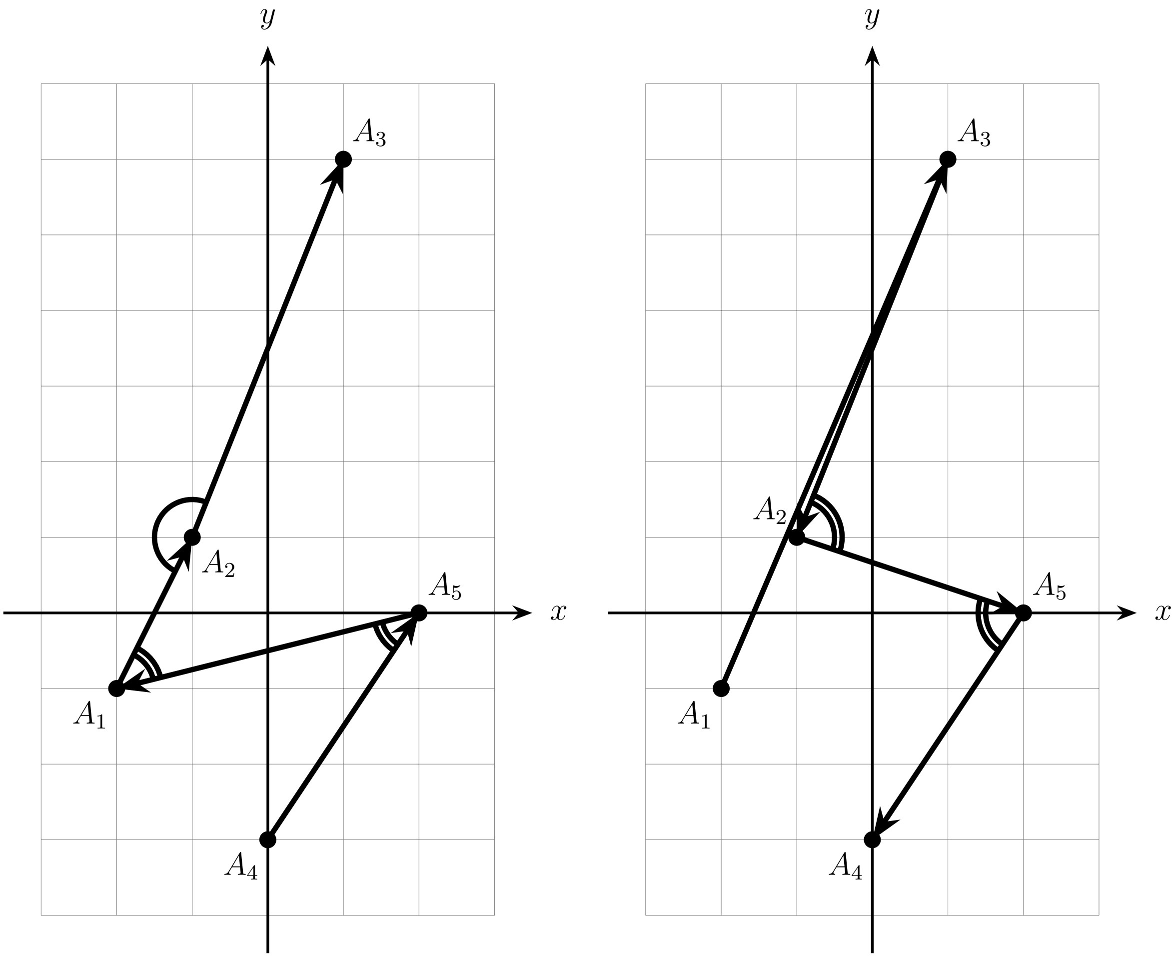
Third example





In the third example the lines have sharpness 2, 3, 4.

Forth example



In the forth example we build lines that have sharpness 2 and 3. The lines have hashes equal to the ones provided earlier.

Problem Sherlock Holmes. Evidence Board

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

Volodya dreams of becoming a detective. Therefore, Volodya often reads books that tell incredible stories of solving crimes. Studying the next case, Volodya came across interesting details of the investigation.

There were a total of n suspected persons in the case. The evidence board contains all n persons. Initially, there were no connections between them.

During the investigation, new connections between suspected persons emerged one after another. Each connection linked two persons that previously had no connection with each other, even indirectly through several other persons.

Let's consider what happened when a connection between persons A and B emerged. In addition to the names of the persons, each connection had three parameters: c_A — the strength of the evidence against A, c_B — the strength of the evidence against B, and w_{AB} — the total strength of the evidence of connection. For natural reasons, the strength of the evidence of connection could not exceed the sum of strengths of the evidence against A and B. That means that for each connection, it was **necessarily** that $w_{AB} \le c_A + c_B$. Upon receiving such connection, the detectives drew a line on the board between the images of persons A and B, assigning the w_{AB} to this line. Also, a sticker with the number c_A was placed on the image of person A, and a sticker with the number c_B was placed on B. If there were already other stickers on the image, the new sticker was placed on top of the old ones.

The case was solved exactly at the moment when all the suspected persons were linked through $n-1$ connections. After solving the crime, the board was placed in the museum in its original form.

Inspired by this approach, Volodya visited that museum and studied the evidence board in detail. Volodya noticed that the image of person v contained stickers with numbers $c_{v,1}, \ldots, c_{v,deg_v}$ numbered **from top to bottom**. Here, deg_v denotes the number of connections associated with person v. Also, Volodya remembered that the i-th connection was between persons a_i and b_i and had evidence strength w_i. Unfortunately the connections were arbitrarily numbered, and their numbers did not necessarily correspond to the order in which they appeared during the investigation.

Due to the confusion with the numbers of connections, the information on the board did not help to restore the process of the investigation. Now Volodya needs to restore any possible chronological order in which the connections could have emerged for the detectives. This task is too difficult for him, so he asks your help. It is also possible that the museum falsified information, and a suitable order does not exist.

Input

The first line of the input contains two integers n and g ($2 \le n \le 200\,000$, $0 \le g \le 9$) — the number of suspected persons in the case and the test group number.

The next $n-1$ lines describe the connections. The i-th line contains three integers a_i, b_i, and w_i ($1 \le a_i, b_i \le n$, $1 \le w_i \le 10^9$, $a_i \ne b_i$) — the persons connected by the i-th connection and the total strength of the i-th connection. It is guaranteed that connections link all persons together.

The next n lines describe the numbers written on the stickers. The i-th line contains deg_i integers $c_{i,1}, \ldots, c_{i,deg_i}$ ($0 \le c_{i,j} \le 10^9$) — the numbers written on the stickers on the image of the i-th person from top to bottom. deg_i equals the number of connections associated with person i.

Output

If there is no suitable chronological order for the restoration of connections according to the conditions of the problem, output "`No`" (without quotes) on a single line.

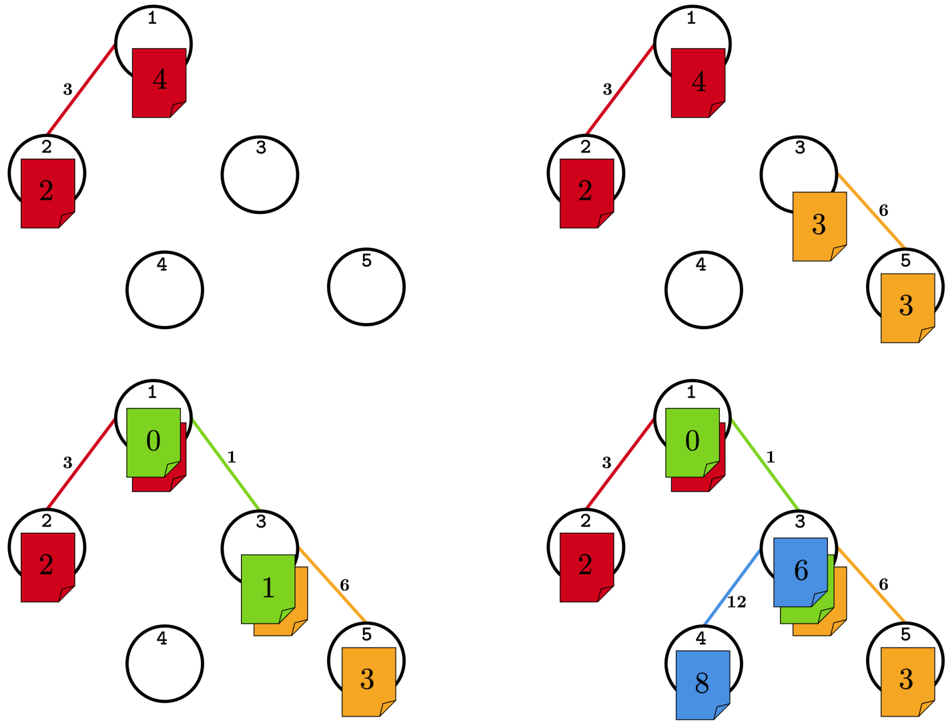
Otherwise, on the first line output "`Yes`" (without quotes). On the second line, output $n-1$ numbers — a suitable chronological order of connections to emerge. The connections are numbered from 1 to $n-1$ in the same order as they are given in the input. If there are multiple possible orders, output any of them.

Examples

input	output
5 0 1 2 3 1 3 1 3 4 12 3 5 6 0 4 2 6 1 3 8 3	Yes 1 4 2 3
7 0 1 2 4 2 3 4 3 4 4 4 5 4 5 6 4 6 7 4 2 1 2 2 3 1 2 3 2 1 2 179	Yes 5 1 2 3 6 4
4 0 1 2 7 1 3 6 1 4 5 3 2 1 5 4 3	No

Note

In the first example, one of the possible orders is $[1, 4, 2, 3]$. In chronological order, the first connection links $A = 1$ and $B = 2$, $c_A = 4, c_B = 2, w_{AB} = 3$, $3 \leq 2 + 4$ — the evidence is correct. The second connection links $A = 3$ and $B = 5$, $c_A = 3, c_B = 3, w_{AB} = 6$, $6 \leq 3 + 3$ — the evidence is correct. The third connection links $A = 1$ and $B = 3$, $c_A = 0, c_B = 1, w_{AB} = 1$, $1 \leq 0 + 1$ — the evidence is correct. The fourth connection links $A = 3$ and $B = 4$, $c_A = 6, c_B = 8, w_{AB} = 12$, $12 \leq 6 + 8$ – the evidence is correct. For a better understanding, refer to the illustration.



Scoring

The tests for this problem consist of nine groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints		Required Groups	Comment
		n	a_i, b_i, c_i, w_i		
0	0	–	–	–	Examples.
1	10	$n \le 10$	–	0	–
2	15	–	$a_i = i, b_i = i+1$ for all i	–	–
3	8	–	$a_i = 1, b_i = i+1$ for all i	–	–
4	9	–	$a_i \le 2, b_i = i+1$ for all i	3	–
5	7	$n \le 1000$	$c_{i,1} \le c_{i,2} \le \ldots \le c_{i,deg_i}$ for all i	–	–
6	7	$n \le 1000$	$c_{i,j} = 0$ for all $1 \le i \le n$ and $j \ge 2$	–	–
7	17	–	$\sum_{v=1}^{n} \sum_{i=1}^{deg_v} c_{v,i} = \sum_{i=1}^{n-1} w_i$	–	–
8	16	$n \le 1000$	–	0, 1, 5, 6	–
9	11	–	–	0 – 8	**Offline-evaluation**

Problem Ded Moroz. More Gifts

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

The organizers of the Closed Olympiad in Informatics decided to prepare gifts for the participants of the Olympiad. A total of k same gift boxes were prepared, each box contains a stack of n gifts. At the top of each stack there is a gift of type a_1, below it there is a gift of type a_2, and so on, at the bottom of the stack there is a gift of type a_n.

The distribution of gifts will be as follows: at the beginning, gifts from the first stack will be given out from top to bottom. After there are no more gifts left in the first stack, gifts from the second stack will be given from top to bottom, and so on, in the end gifts from the k-th stack will be given.

A participant can receive several gifts at once, so at the beginning gifts will be given to the first participant, then to the second, and so on. It is known that if a participant receives more than t different types of gifts, the participant will be too happy and will write the Olympiad poorly. In order for the participants to write the Olympiad well, it was decided to give each participant no more than t different types of gifts (note that a participant may receive several gifts of the same type).

The organizers of the Closed Olympiad in Informatics decided to make the Olympiad exclusive and invite as few participants as possible. Help the organizers find out the minimum number of participants they can invite so that all the gifts are distributed to the participants, and each participant receives no more than t different types of gifts.

Input

The first line of the input contains three integers n, k, and t ($1 \le n \le 300\,000$, $1 \le k, t \le 10^9$) — the number of gifts in one stack, the number of stacks of gifts, and the maximum number of different types of gifts that can be received by one participant.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the types of gifts, in the order from top to bottom of the stack.

Output

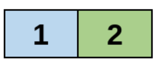
Output a single number — the minimum number of participants, such that all the gifts will be distributed to them, and each participant receives no more than t different types of gifts.

Examples

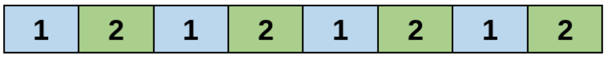
input	output
`2 4 1` `1 2`	`8`
`4 3 1` `1 1 2 1`	`7`
`7 2 3` `1 2 3 4 5 6 7`	`5`

Note

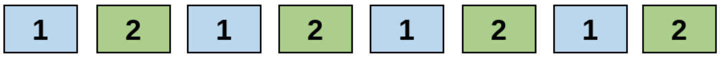
In the first example, the stack contains the following types of gifts (in order from top to bottom). Different colors denote different positions in the stack.



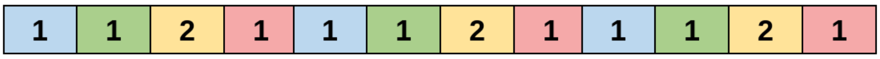
There are a total of 4 stacks of gifts, so the gifts will be given out in the following order:

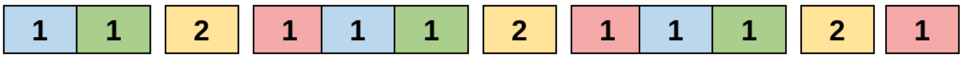


Since $t = 1$, each participant in this case can only receive gifts of one type:

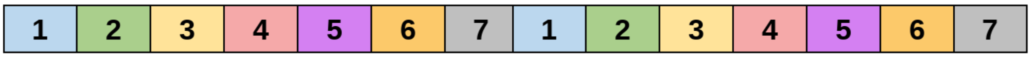


In the second example, the order of gift distribution and the final sets of gifts are following:

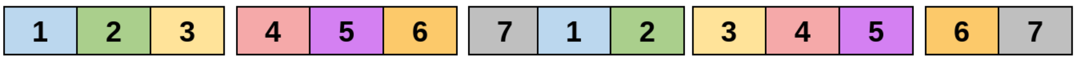




In the third example, the order of gift distribution is as follows:



In this case, one of the possible optimal distribution of gifts into sets is the following:



Scoring

The tests for this problem consist of six groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Note that passing the example tests is not required for some groups.

Group	Points	Additional constraints			Required groups	Comment
		n	k	t		
0	0	–	–	–	–	Examples.
1	14	$n \le 100$	$k \le 10$	–	0	–
2	12	–	–	$t = 1$	–	–
3	16	$n \le 1000$	$k \le 1000$	–	0, 1	–
4	21	$n \le 1500$	$k \le 10^6$	–	0, 1, 3	–
5	18	–	$k \le 10^6$	–	0, 1, 3, 4	–
6	19	–	–	–	0 – 5	–

Problem Scheherazade. Big Persimmon

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	1024 megabytes

Alice and Bob bought a big persimmon, cut it into n pieces with sizes $w_1, \ldots, w_n$, and immediately started eating it. The kids will eat the pieces simultaneously, and for each of them, the eating process is as follows:

As soon as someone finishes eating their previous piece (and at the beginning of the meal), they choose the next piece and start eating it. If a piece of size w is taken, it will take exactly w seconds to eat it, and then it will be time to choose a new piece. If both finish eating their previous piece at the same time (or if the eating just started), Alice will choose the first piece, but they will start eating at the same time. Choosing a new piece does not take time.

Since both Alice and Bob are perfectionists, when they choose a piece, they will take either the smallest (with the smallest w_i) or the largest (with the largest w_i) from all the remaining pieces.

The eating process ends when the last person finishes eating and there are no more pieces left.

Both Alice and Bob are interested in eating as much as possible. Find the total size of the pieces that Alice will eat and the total size of the pieces that Bob will eat, if they both choose the pieces optimally.

Input

The first line contains a single integer n ($1 \le n \le 2000$) — the number of persimmon pieces.

The second line contains n integers $w_1, w_2, \ldots, w_n$ ($1 \le w_i \le 20\,000$, $w_i \le w_{i+1}$) — the sizes of the persimmon pieces.

Let W denote the sum of the sizes of all the pieces. It is guaranteed that $W \le 20\,000$.

Output

In a single line, output two numbers — the total size of the pieces that Alice will eat and the total size of the pieces that Bob will eat, if they both choose the pieces optimally.

Examples

input	output
5 1 1 3 4 6	8 7
4 1 1 2 2	3 3
4 1 7 7 9	10 14

Note

In the first example, Alice should first take a piece of size 1. Immediately after that, Bob should also take a piece of size 1. After a second, Alice will take a piece of size 3, and then Bob will take a piece of size 6. 3 seconds later, Alice will take a piece of size 4. 3 seconds later, Bob will finish eating, and a second later the process will finish. At this point, Alice will eat pieces of sizes $1 + 3 + 4 = 8$, and Bob: $1 + 6 = 7$.

In the third example, Alice should take a piece of size 1, and Bob should take a piece of size 7. After a second, Alice will take a piece of size 9, and 6 seconds later, Bob will take a piece of size 9.

Scoring

The tests for this problem consist of four groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed.. Note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the competition.

Group	Points	Additional constraints		Required Groups	Comment
		n	w_i		
0	0	–	–	–	Examples.
1	10	$n = 3$	–	–	–
2	12	–	$w_i \leq 2$	–	–
3	19	$n \leq 200$	$w_i \leq 500$	0	–
4	15	$n \leq 500$	$W \leq 5000$	–	$w_{i+1} \leq 2 \cdot w_i$ for all $1 \leq i \leq n-1$
5	13	–	–	2, 4	$w_{i+1} \leq 2 \cdot w_i$ for all $1 \leq i \leq n-1$
6	31	–	–	$0-5$	**Offline-evaluation.**