VK Cup 2016 - Round 1 (Div. 2 Edition)

A. Bear and Reverse Radewoosh

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

Limak and Radewoosh are going to compete against each other in the upcoming algorithmic contest. They are equally skilled but they won't solve problems in the same order.

There will be n problems. The i-th problem has initial score p_i and it takes exactly t_i minutes to solve it. Problems are sorted by difficulty — it's guaranteed that $p_i < p_{i+1}$ and $t_i < t_{i+1}$.

A constant c is given too, representing the speed of loosing points. Then, submitting the i-th problem at time x (x minutes after the start of the contest) gives $max(0,\ p_i - c \cdot x)$ points.

Limak is going to solve problems in order $1, 2, ..., n$ (sorted increasingly by p_i). Radewoosh is going to solve them in order $n, n - 1, ..., 1$ (sorted decreasingly by p_i). Your task is to predict the outcome — print the name of the winner (person who gets more points at the end) or a word "`Tie`" in case of a tie.

You may assume that the duration of the competition is greater or equal than the sum of all t_i. That means both Limak and Radewoosh will accept all n problems.

Input

The first line contains two integers n and c ($1 \le n \le 50,\ 1 \le c \le 1000$) — the number of problems and the constant representing the speed of loosing points.

The second line contains n integers $p_1, p_2, ..., p_n$ ($1 \le p_i \le 1000, p_i < p_{i+1}$) — initial scores.

The third line contains n integers $t_1, t_2, ..., t_n$ ($1 \le t_i \le 1000, t_i < t_{i+1}$) where t_i denotes the number of minutes one needs to solve the i-th problem.

Output

Print "`Limak`" (without quotes) if Limak will get more points in total. Print "`Radewoosh`" (without quotes) if Radewoosh will get more points in total. Print "`Tie`" (without quotes) if Limak and Radewoosh will get the same total number of points.

Examples

input
3 2 50 85 250 10 15 25
output
Limak

input
3 6 50 85 250 10 15 25
output
Radewoosh

input
8 1 10 20 30 40 50 60 70 80 8 10 58 63 71 72 75 76
output
Tie

Note

In the first sample, there are 3 problems. Limak solves them as follows:

1. Limak spends 10 minutes on the 1-st problem and he gets $50 - c \cdot 10 = 50 - 2 \cdot 10 = 30$ points.
2. Limak spends 15 minutes on the 2-nd problem so he submits it $10 + 15 = 25$ minutes after the start of the contest. For the 2-nd problem he gets $85 - 2 \cdot 25 = 35$ points.

3. He spends 25 minutes on the 3-rd problem so he submits it $10 + 15 + 25 = 50$ minutes after the start. For this problem he gets $250 - 2 \cdot 50 = 150$ points.

So, Limak got $30 + 35 + 150 = 215$ points.

Radewoosh solves problem in the reversed order:

1. Radewoosh solves 3-rd problem after 25 minutes so he gets $250 - 2 \cdot 25 = 200$ points.
2. He spends 15 minutes on the 2-nd problem so he submits it $25 + 15 = 40$ minutes after the start. He gets $85 - 2 \cdot 40 = 5$ points for this problem.
3. He spends 10 minutes on the 1-st problem so he submits it $25 + 15 + 10 = 50$ minutes after the start. He gets $max(0, 50 - 2 \cdot 50) = max(0, -50) = 0$ points.

Radewoosh got $200 + 5 + 0 = 205$ points in total. Limak has 215 points so Limak wins.

In the second sample, Limak will get 0 points for each problem and Radewoosh will first solve the hardest problem and he will get $250 - 6 \cdot 25 = 100$ points for that. Radewoosh will get 0 points for other two problems but he is the winner anyway.

In the third sample, Limak will get 2 points for the 1-st problem and 2 points for the 2-nd problem. Radewoosh will get 4 points for the 8-th problem. They won't get points for other problems and thus there is a tie because $2 + 2 = 4$.

B. Bear and Displayed Friends

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

Limak is a little polar bear. He loves connecting with other bears via social networks. He has n friends and his relation with the i-th of them is described by a unique integer t_i. The bigger this value is, the better the friendship is. No two friends have the same value t_i.

Spring is starting and the Winter sleep is over for bears. Limak has just woken up and logged in. All his friends still sleep and thus none of them is online. Some (maybe all) of them will appear online in the next hours, one at a time.

The system displays friends who are online. On the screen there is space to display at most k friends. If there are more than k friends online then the system displays only k best of them — those with biggest t_i.

Your task is to handle queries of two types:

- "`1 id`" — Friend id becomes online. It's guaranteed that he wasn't online before.
- "`2 id`" — Check whether friend id is displayed by the system. Print "`YES`" or "`NO`" in a separate line.

Are you able to help Limak and answer all queries of the second type?

Input

The first line contains three integers n, k and q ($1 \le n, q \le 150\,000$, $1 \le k \le min(6, n)$) — the number of friends, the maximum number of displayed online friends and the number of queries, respectively.

The second line contains n integers $t_1, t_2, ..., t_n$ ($1 \le t_i \le 10^9$) where t_i describes how good is Limak's relation with the i-th friend.

The i-th of the following q lines contains two integers $type_i$ and id_i ($1 \le type_i \le 2$, $1 \le id_i \le n$) — the i-th query. If $type_i = 1$ then a friend id_i becomes online. If $type_i = 2$ then you should check whether a friend id_i is displayed.

It's guaranteed that no two queries of the first type will have the same id_i becuase one friend can't become online twice. Also, it's guaranteed that at least one query will be of the second type ($type_i = 2$) so the output won't be empty.

Output

For each query of the second type print one line with the answer — "`YES`" (without quotes) if the given friend is displayed and "`NO`" (without quotes) otherwise.

Examples

input

```
4 2 8
300 950 500 200
1 3
2 4
2 3
1 1
1 2
2 1
2 2
2 3
```

output

```
NO
YES
NO
YES
YES
```

input

```
6 3 9
50 20 51 17 99 24
1 3
1 4
1 5
1 2
2 4
2 2
1 1
2 4
2 3
```

output

```
NO
YES
NO
YES
```

Note

In the first sample, Limak has 4 friends who all sleep initially. At first, the system displays nobody because nobody is online. There are the following 8 queries:

1. "1 3" — Friend 3 becomes online.
2. "2 4" — We should check if friend 4 is displayed. He isn't even online and thus we print "NO".
3. "2 3" — We should check if friend 3 is displayed. Right now he is the only friend online and the system displays him. We should print "YES".
4. "1 1" — Friend 1 becomes online. The system now displays both friend 1 and friend 3.
5. "1 2" — Friend 2 becomes online. There are 3 friends online now but we were given $k = 2$ so only two friends can be displayed. Limak has worse relation with friend 1 than with other two online friends ($t_1 < t_2, t_3$) so friend 1 won't be displayed
6. "2 1" — Print "NO".
7. "2 2" — Print "YES".
8. "2 3" — Print "YES".

C. Bear and Forgotten Tree 3

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

A tree is a connected undirected graph consisting of n vertices and $n - 1$ edges. Vertices are numbered 1 through n.

Limak is a little polar bear and Radewoosh is his evil enemy. Limak once had a tree but Radewoosh stolen it. Bear is very sad now because he doesn't remember much about the tree — he can tell you only three values n, d and h:

- The tree had exactly n vertices.
- The tree had diameter d. In other words, d was the biggest distance between two vertices.
- Limak also remembers that he once rooted the tree in vertex 1 and after that its height was h. In other words, h was the biggest distance between vertex 1 and some other vertex.

The distance between two vertices of the tree is the number of edges on the simple path between them.

Help Limak to restore his tree. Check whether there exists a tree satisfying the given conditions. Find any such tree and print its edges in any order. It's also possible that Limak made a mistake and there is no suitable tree – in this case print "-1".

Input

The first line contains three integers n, d and h ($2 \le n \le 100\,000$, $1 \le h \le d \le n - 1$) — the number of vertices, diameter, and height after rooting in vertex 1, respectively.

Output

If there is no tree matching what Limak remembers, print the only line with "-1" (without the quotes).

Otherwise, describe any tree matching Limak's description. Print $n - 1$ lines, each with two space-separated integers – indices of vertices connected by an edge. If there are many valid trees, print any of them. You can print edges in any order.

Examples

input
5 3 2
output
1 2 1 3 3 4 3 5

input
8 5 2
output
-1

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

Note

Below you can see trees printed to the output in the first sample and the third sample.

D. Bear and Polynomials

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

Limak is a little polar bear. He doesn't have many toys and thus he often plays with polynomials.

He considers a polynomial *valid* if its degree is n and its coefficients are integers not exceeding k by the absolute value. More formally:

Let $a_0, a_1, ..., a_n$ denote the coefficients, so . Then, a polynomial $P(x)$ is valid if all the following conditions are satisfied:

- a_i is integer for every i;
- $|a_i| \leq k$ for every i;
- $a_n \neq 0$.

Limak has recently got a valid polynomial P with coefficients $a_0, a_1, a_2, ..., a_n$. He noticed that $P(2) \neq 0$ and he wants to change it. He is going to change one coefficient to get a **valid** polynomial Q of degree n that $Q(2) = 0$. Count the number of ways to do so. You should count two ways as a distinct if coefficients of target polynoms differ.

Input

The first line contains two integers n and k $(1 \leq n \leq 200\,000, 1 \leq k \leq 10^9)$ — the degree of the polynomial and the limit for absolute values of coefficients.

The second line contains $n + 1$ integers $a_0, a_1, ..., a_n$ $(|a_i| \leq k, a_n \neq 0)$ — describing a **valid** polynomial . It's guaranteed that $P(2) \neq 0$.

Output

Print the number of ways to change one coefficient to get a valid polynomial Q that $Q(2) = 0$.

Examples

input
3 1000000000 10 -9 -3 5
output
3

input
3 12 10 -9 -3 5
output
2

input
2 20 14 -7 19
output
0

Note

In the first sample, we are given a polynomial $P(x) = 10 - 9x - 3x^2 + 5x^3$.

Limak can change one coefficient in three ways:

1. He can set $a_0 = -10$. Then he would get $Q(x) = -10 - 9x - 3x^2 + 5x^3$ and indeed $Q(2) = -10 - 18 - 12 + 40 = 0$.
2. Or he can set $a_2 = -8$. Then $Q(x) = 10 - 9x - 8x^2 + 5x^3$ and indeed $Q(2) = 10 - 18 - 32 + 40 = 0$.
3. Or he can set $a_1 = -19$. Then $Q(x) = 10 - 19x - 3x^2 + 5x^3$ and indeed $Q(2) = 10 - 38 - 12 + 40 = 0$.

In the second sample, we are given the same polynomial. This time though, k is equal to 12 instead of 10^9. Two first of ways listed above are still valid but in the third way we would get $|a_1| > k$ what is not allowed. Thus, the answer is 2 this time.

E. Bear and Contribution

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Codeforces is a wonderful platform and one its feature shows how much someone contributes to the community. Every registered user has *contribution* — an integer number, not necessarily positive. There are n registered users and the i-th of them has contribution t_i.

Limak is a little polar bear and he's new into competitive programming. He doesn't even have an account in Codeforces but he is able to upvote existing blogs and comments. We assume that every registered user has infinitely many blogs and comments.

- Limak can spend b minutes to read one blog and upvote it. Author's contribution will be increased by 5.
- Limak can spend c minutes to read one comment and upvote it. Author's contribution will be increased by 1.

Note that it's possible that Limak reads blogs faster than comments.

Limak likes ties. He thinks it would be awesome to see a tie between at least k registered users. To make it happen he is going to spend some time on reading and upvoting. After that, there should exist an integer value x that at least k registered users have contribution exactly x.

How much time does Limak need to achieve his goal?

Input

The first line contains four integers n, k, b and c ($2 \le k \le n \le 200\,000$, $1 \le b, c \le 1000$) — the number of registered users, the required minimum number of users with the same contribution, time needed to read and upvote a blog, and time needed to read and upvote a comment, respectively.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($|t_i| \le 10^9$) where t_i denotes contribution of the i-th registered user.

Output

Print the minimum number of minutes Limak will spend to get a tie between at least k registered users.

Examples

input
4 3 100 30 12 2 6 1
output
220

input
4 3 30 100 12 2 6 1
output
190

input
6 2 987 789 -8 42 -4 -65 -8 -8
output
0

Note

In the first sample, there are 4 registered users and Limak wants a tie between at least 3 of them. Limak should behave as follows.

- He spends 100 minutes to read one blog of the 4-th user and increase his contribution from 1 to 6.
- Then he spends $4 \cdot 30 = 120$ minutes to read four comments of the 2-nd user and increase his contribution from 2 to 6 (four times it was increaded by 1).

In the given scenario, Limak spends $100 + 4 \cdot 30 = 220$ minutes and after that each of users $2, 3, 4$ has contribution 6.

In the second sample, Limak needs 30 minutes to read a blog and 100 minutes to read a comment. This time he can get 3 users with contribution equal to 12 by spending $100 + 3 \cdot 30 = 190$ minutes:

- Spend $2 \cdot 30 = 60$ minutes to read two blogs of the 1-st user to increase his contribution from 2 to 12.
- Spend $30 + 100$ minutes to read one blog and one comment of the 3-rd user. His contribution will change from 6 to $6 + 5 + 1 = 12$.