Footwork

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	1 second	256 MiB

The corridor can be thought of as a grid of squares containing two rows and N columns (numbered from 1 to N from left to right). Each square contains an integer value, which can be *negative*:

- The i-th element (counting from 1) in the top row is A_i.
- The i-th element (counting from 1) in the bottom row is B_i.

You are a human with two feet. One foot starts on the square A_1 and the other foot starts on the square B_1. You must move your feet so that one foot ends on the square A_N and the other foot ends on the square B_N.

You move your feet by making *steps*: In each step, you pick one foot and move it to another square to the right in the same row. After each step, your feet must be at most K squares away from each other. More formally, if one foot is on square A_i, and the other foot is on square B_j, then $|i - j| \leq K$ must hold.

You *may* make multiple steps in a row with the same foot.

At the end, your *score* is the sum of values of all the squares you stepped on (including the starting and ending squares). What is the maximum score possible?

Subtasks and Constraints

For all subtasks, you are guaranteed that:

- $1 \leq N \leq 100\,000$.
- $1 \leq K \leq 100\,000$.
- $-10\,000 \leq A_i \leq 10\,000$.
- $-10\,000 \leq B_i \leq 10\,000$.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	12	$K \leq 5$
2	24	$A_i = 0$ or -1, for all i. $B_i = 0$ or -1, for all i.
3	6	$N \leq 300$
4	20	$N \leq 3\,000$
5	38	No further constraints apply.

Input

- The first line of input contains the two integers, N and K.
- The second line contains N integers. The i-th integer (starting from 1) is A_i.
- The third line contains N integers. The i-th integer (starting from 1) is B_i.

Output

The output should contain a single integer: the maximum score possible.

Sample Input 1

```
4 1
0 2 2 8
0 -10 5 2
```

Sample Output 1

```
19
```

Sample Input 2

```
7 2
0 -10 -6 2 -10 0 0
5 3 -2 -1 -10 -10 0
```

Sample Output 2

```
9
```

Explanation

In Sample Case 1, your score is $0 + 2 + 2 + 8 + 0 + 5 + 2 = 19$.

In Sample Case 2, your score is $0 + 2 + 0 + 0 + 5 + 3 + -1 + 0 = 9$.

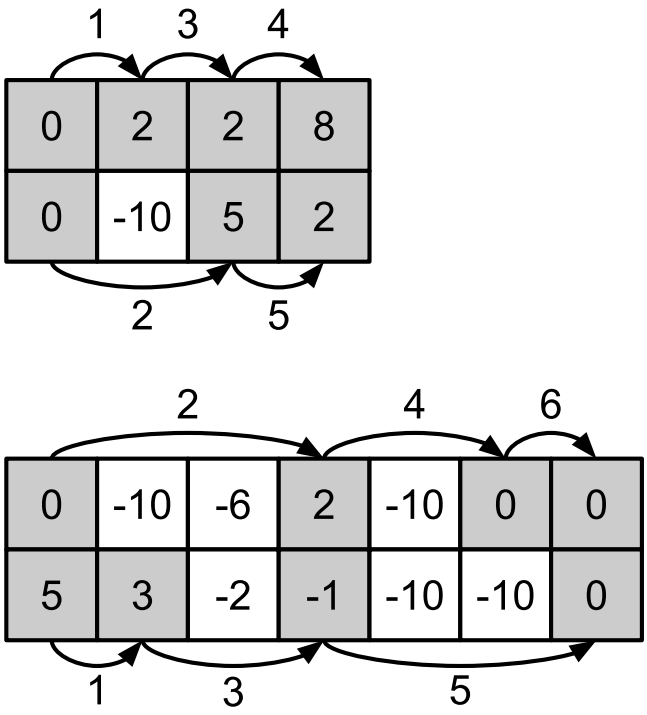

Figure 1: The steps are numbered in the order you shoud make them. The shaded squares are the ones you stepped on.

Loss of Balance

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	2 seconds	256 MiB

An array A is *(x, y)-fair*, if there is an index i and an index j such that:

- $i < j$, and
- $A_i = x$ and $A_j = y$.

An array A is *K-balanced* if and only if:

- All elements of the array are integers between 1 and K.
- Every integer from 1 to K appears *at least once* in A.
- A is (x, y)-fair and (y, x)-fair for all pairs of integers x and y (from 1 to K) where $x \neq y$.

For example:

- `[1, 2, 3, 4]` is not 3-balanced, since the array contains a `4`.
- `[1, 3]` is not 3-balanced, since the array does not contain a `2`.
- `[1, 2, 1, 3, 1]` is not 3 balanced, since it is not (3, 2)-fair.
- `[1, 2, 3, 2, 1]` is 3-balanced.

Hugo has an array A containing N elements that is K-balanced. Unfortunately, he lost the array and would like you to help him recover it.

Fortunately, Hugo recalls a relative ordering of the elements of A. More precisely, he has an array B also of length N. For all i and j:

- if $B_i = B_j$, then $A_i = A_j$,
- if $B_i > B_j$, then $A_i \geq A_j$ (**large** inequality), and
- if $B_i < B_j$, then $A_i \leq A_j$ (**large** inequality).

For example:

- If $B =$ `[1, 5, 2]`, then $A_1 \leq A_3 \leq A_2$.
- If $B =$ `[6, 3, 7, 7, 3, 7]`, then $A_3 = A_4 = A_6$ and $A_2 = A_5$. Furthermore, $A_5 \leq A_1 \leq A_6$.

Please help Hugo recover A, or say that it is impossible!

Subtasks and Constraints

For all subtasks, you are guaranteed that:

- $2 \leq N \leq 200\,000$.
- $2 \leq K \leq N$.
- $1 \leq B_i \leq 200\,000$, for all i.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	10	$N \leq 10$ and $B_i \leq K$, for all i.
2	20	$B_i \leq K$, for all i.
3	30	$N \leq 3000$
4	10	$K \leq 5$
5	30	No further constraints apply.

Input

- The first line of input contains the two integers, N and K.
- The second line contains N integers. The i-th integer (starting from 1) is B_i.

Output

The output should contain N integers on a single line. The i-th of these integers should be A_i, and each integer **must** be between 1 and K.

The printed array A must be K-balanced **and** must match the relative order defined by B.

If there are many such correct array, you can output **any of them**. If no such array exists, print `-1` instead.

Sample Input 1

```
5 3
5 300 900 300 6
```

Sample Output 1

```
1 2 3 2 1
```

Sample Input 2

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

Sample Output 2

```
-1
```

Sample Input 3

```
6 2
1 2 2 1 2 1
```

Sample Output 3

```
1 2 2 1 2 1
```

Explanation

In Sample Input 1, the **only** correct output is `[1, 2, 3, 2, 1]` (it is 3-balanced **and** it matches the relative order of B).

Note that, for example, the three following arrays would be **incorrect** output:

- `[3, 1, 2, 3, 1]` is 3-balanced, but does not match the relative order of B
- `[1, 2, 3, 2, 2]` matches the relative order of B, but is not 3-balanced.
- `[1, 1, 2, 1, 1]` matches the relative order of B, but is not 3-balanced.

In Sample Input 2, it is impossible to create an increasing 5-balanced array.

In Sample Input 3, the only correct output is `[1, 2, 2, 1, 2, 1]`. Note that Sample Input 3 satisfies the constraints of all subtasks.

Emergency Reinforcement

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	5 seconds	256 MiB

There are N islands (numbered from 1 to N) connected by E two-way bridges (numbered from 1 to E). The bridges were built by one of K companies (numbered from 1 to K).

The i-th bridge connects island A_i and B_i, and was built by company C_i. The same pair of islands could be connected by more than one bridge. No bridge connects an island to itself.

Very soon, a large earthquake will hit the islands and destroy all of the bridges! Each company has X dollars that they can spend to *reinforce* bridges that they built, saving them from being destroyed by the earthquake. To reinforce the i-th bridge, company C_i must spend D_i dollars.

The companies would like to minimise the total number of connected components after the earthquake. Two islands are in the same connected component if and only if there exists a way to travel between them using only reinforced bridges. Can you help them?

Note: please read the Scoring section below.

Subtasks and Constraints

For all subtasks, you are guaranteed that:

- $1 \leq N \leq 10\,000$.
- $1 \leq E \leq 100\,000$.
- $1 \leq K \leq 5\,000$.
- $1 \leq X \leq 1\,000\,000\,000$.
- $1 \leq A_i \leq N$.
- $1 \leq B_i \leq N$.
- $A_i \neq B_i$, for all i.
- $1 \leq C_i \leq K$.
- $1 \leq D_i \leq X$.

In this problem, each subtask **only has one test case**. These test cases are available for download from the Attachments page.

Subtask	Points	Additional constraints
1	5	$N = 7$ and $E = 8$.
2	5	Between any two islands there is a unique path (sequence of bridges).
3	15	$K = 1$.
4	15	$B_i = N$, for all i. If $A_i = A_j$ then $D_i = D_j$, for all i, j.
5	15	$D_i = 1$, for all i. Each island is connected to at most two islands.
6	15	$D_i = 1$, for all i.
7	15	-
8	15	-

You are encouraged to look at the content of the test cases, and to experiment on your computer. Submitting a code which prints an hardcoded solution for one of the test cases **is** allowed.

Input

- The first line of input contains the four integers, N, E, K and X.
- Then, E lines follow. The i-th line contains the four integers A_i, B_i, C_i and D_i.

Output

Output a single line, containing up to E integers, the bridges that you would like to reinforce (in any order).

Scoring

If you:

- list the same bridge more than once, or
- output a number less than 1 or more than E, or
- spend too many dollars of any company,

then your score will be zero for that subtask.

Otherwise, your score will be a sliding scale based on how close your solution is to the optimal solution. Given two parameters INF and SUP, if the number of connected component after the earthquake is X, your score on this subtask will be:

$$min(100, max(0, 100 * (SUP - X)/(SUP - INF)))$$

Scoring parameters of each subtask are given in the table below:

Subtask	INF	SUP
1	1	4
2	382	1000
3	58	176
4	1	3
5	151	1000
6	1	1000
7	1	150
8	1	1100

Sample Input 1

```
10 11 8 1000
1 2 7 100
1 3 7 100
2 3 4 750
2 3 1 1000
4 3 4 750
4 5 3 600
5 6 3 601
5 6 3 602
6 4 3 603
4 6 3 604
7 8 3 100
```

Sample Output 1

```
2 4 5 7 11
```

Sample Input 2

```
4 5 5 12345
4 1 1 12345
1 2 2 12345
2 4 5 12345
4 3 2 12345
3 2 3 12345
```

Sample Output 2

```
1 2 3 5
```

Explanation

In Sample Case 1, each company has $X = 1000$ dollars to spend:

- Company 1 reinforces the 4th bridge, costing 1000 dollars.
- Company 3 reinforces the 7th and 11th bridges, costing $601 + 100 = 701$ dollars.
- Company 4 reinforces the 5th bridge, costing 750 dollars.
- Company 7 reinforces the 2nd bridge, costing 100 dollars.

This gives 5 connected components (two of those components are size 1).

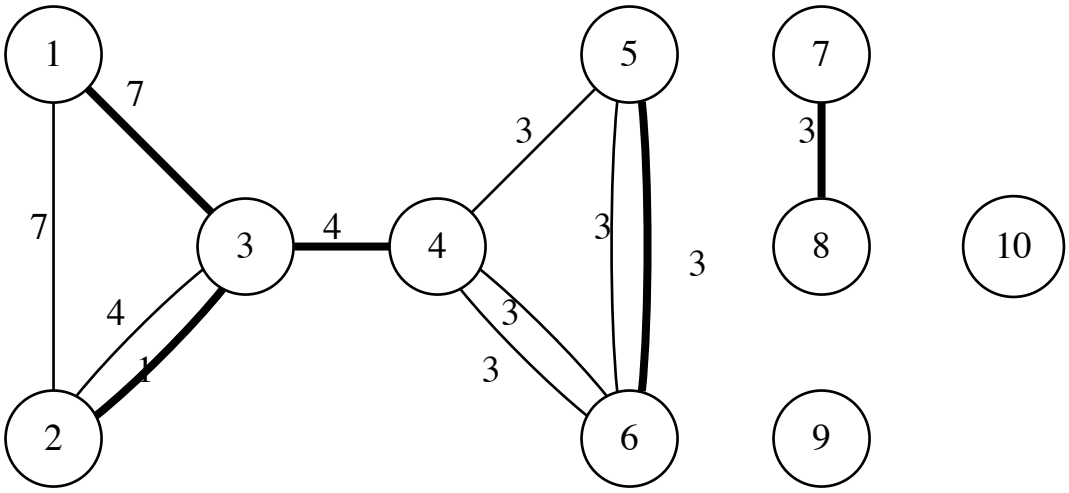

Figure 1: Sample Case 1

In Sample Case 2, each company has $X = 12345$ dollars to spend:

- Company 1 reinforces the 1st bridge, costing 12345 dollars.
- Company 2 reinforces the 2nd bridge, costing 12345 dollars.
- Company 3 reinforces the 5th bridge, costing 12345 dollars.
- Company 5 reinforces the 3rd bridge, costing 12345 dollars.

This gives 1 connected component.

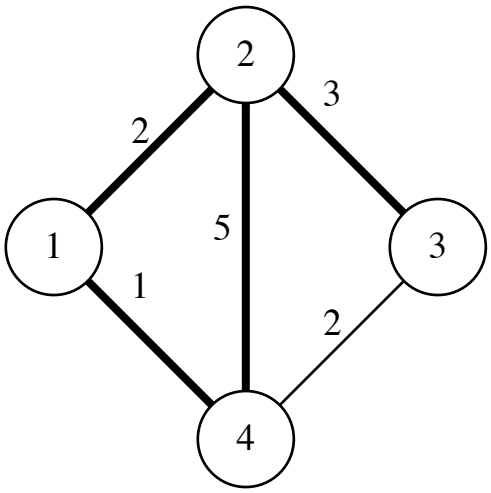

Figure 2: Sample Case 2