CODEFORCES β
Sponsored by Telegram

Codeforces Round #374 (Div. 2)

A. One-dimensional Japanese Crossword

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

Recently Adaltik discovered japanese crosswords. Japanese crossword is a picture, represented as a table sized $a \times b$ squares, and each square is colored white or black. There are integers to the left of the rows and to the top of the columns, encrypting the corresponding row or column. The number of integers represents how many groups of black squares there are in corresponding row or column, and the integers themselves represents the number of consecutive black squares in corresponding group (you can find more detailed explanation in Wikipedia https://en.wikipedia.org/wiki/Japanese_crossword).

Adaltik decided that the general case of japanese crossword is too complicated and drew a row consisting of n squares (e.g. japanese crossword sized $1 \times n$), which he wants to encrypt in the same way as in japanese crossword.

The example of encrypting of a single row of japanese crossword.

Help Adaltik find the numbers encrypting the row he drew.

Input

The first line of the input contains a single integer n $(1 \le n \le 100)$ — the length of the row. The second line of the input contains a single string consisting of n characters 'B' or 'W', ('B' corresponds to black square, 'W' — to white square in the row that Adaltik drew).

Output

The first line should contain a single integer k — the number of integers encrypting the row, e.g. the number of groups of black squares in the row.

The second line should contain k integers, encrypting the row, e.g. corresponding to sizes of groups of consecutive black squares in the order from left to right.

Examples

input
```
3
BBW
```
output
```
1
2
```

input
```
5
BWBWB
```
output
```
3
1 1 1
```

input
```
4
WWWW
```
output
```
0
```

input
```
4
BBBB
```
output
```
1
4
```

input
```
13
WBBBBWWBWBBBW
```
output

```
3
4 1 3
```

Note

The last sample case correspond to the picture in the statement.

B. Passwords

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

Vanya is managed to enter his favourite site Codehorses. Vanya uses n distinct passwords for sites at all, however he can't remember which one exactly he specified during Codehorses registration.

Vanya will enter passwords in order of non-decreasing their lengths, and he will enter passwords of same length in arbitrary order. Just when Vanya will have entered the correct password, he is immediately authorized on the site. Vanya will not enter any password twice.

Entering any passwords takes one second for Vanya. But if Vanya will enter wrong password k times, then he is able to make the next try only 5 seconds after that. Vanya makes each try immediately, that is, at each moment when Vanya is able to enter password, he is doing that.

Determine how many seconds will Vanya need to enter Codehorses in the best case for him (if he spends minimum possible number of second) and in the worst case (if he spends maximum possible amount of seconds).

Input

The first line of the input contains two integers n and k $(1 \le n, k \le 100)$ — the number of Vanya's passwords and the number of failed tries, after which the access to the site is blocked for 5 seconds.

The next n lines contains passwords, one per line — pairwise distinct non-empty strings consisting of latin letters and digits. Each password length does not exceed 100 characters.

The last line of the input contains the Vanya's Codehorses password. It is guaranteed that the Vanya's Codehorses password is equal to some of his n passwords.

Output

Print two integers — time (in seconds), Vanya needs to be authorized to Codehorses in the best case for him and in the worst case respectively.

Examples

input

```
5 2
cba
abc
bb1
abC
ABC
abc
```

output

```
1 15
```

input

```
4 100
11
22
1
2
22
```

output

```
3 4
```

Note

Consider the first sample case. As soon as all passwords have the same length, Vanya can enter the right password at the first try as well as at the last try. If he enters it at the first try, he spends exactly 1 second. Thus in the best case the answer is 1. If, at the other hand, he enters it at the last try, he enters another 4 passwords before. He spends 2 seconds to enter first 2 passwords, then he waits 5 seconds as soon as he made 2 wrong tries. Then he spends 2 more seconds to enter 2 wrong passwords, again waits 5 seconds and, finally, enters the correct password spending 1 more second. In summary in the worst case he is able to be authorized in 15 seconds.

Consider the second sample case. There is no way of entering passwords and get the access to the site blocked. As soon as the required password has length of 2, Vanya enters all passwords of length 1 anyway, spending 2 seconds for that. Then, in the best case, he immediately enters the correct password and the answer for the best case is 3, but in the worst case he enters wrong password of length 2 and only then the right one, spending 4 seconds at all.

C. Journey

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Irina arrived to one of the most famous cities of Berland — the Berlatov city. There are n showplaces in the city, numbered from 1 to n, and some of them are connected by one-directional roads. The roads in Berlatov are designed in a way such that there **are no** cyclic routes between showplaces.

Initially Irina stands at the showplace 1, and the endpoint of her journey is the showplace n. Naturally, Irina wants to visit as much showplaces as she can during her journey. However, Irina's stay in Berlatov is limited and she can't be there for more than T time units.

Help Irina determine how many showplaces she may visit during her journey from showplace 1 to showplace n within a time not exceeding T. It is guaranteed that there is at least one route from showplace 1 to showplace n such that Irina will spend no more than T time units passing it.

Input

The first line of the input contains three integers n, m and T ($2 \le n \le 5000,\ 1 \le m \le 5000,\ 1 \le T \le 10^9$) — the number of showplaces, the number of roads between them and the time of Irina's stay in Berlatov respectively.

The next m lines describes roads in Berlatov. i-th of them contains 3 integers u_i, v_i, t_i ($1 \le u_i, v_i \le n, u_i \ne v_i, 1 \le t_i \le 10^9$), meaning that there is a road starting from showplace u_i and leading to showplace v_i, and Irina spends t_i time units to pass it. It is guaranteed that the roads do not form cyclic routes.

It is guaranteed, that there is at most one road between each pair of showplaces.

Output

Print the single integer k ($2 \le k \le n$) — the maximum number of showplaces that Irina can visit during her journey from showplace 1 to showplace n within time not exceeding T, in the first line.

Print k distinct integers in the second line — indices of showplaces that Irina will visit on her route, in the order of encountering them.

If there are multiple answers, print any of them.

Examples

input

```
4 3 13
1 2 5
2 3 7
2 4 8
```

output

```
3
1 2 4
```

input

```
6 6 7
1 2 2
1 3 3
3 6 3
2 4 2
4 6 2
6 5 1
```

output

```
4
1 2 4 6
```

input

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

output

```
3
1 3 5
```

D. Maxim and Array

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

Recently Maxim has found an array of n integers, needed by no one. He immediately come up with idea of changing it: he invented positive integer x and decided to add or subtract it from arbitrary array elements. Formally, by applying single operation Maxim chooses integer i $(1 \leq i \leq n)$ and replaces the i-th element of array a_i either with $a_i + x$ or with $a_i - x$. Please note that the operation may be applied more than once to the same position.

Maxim is a curious minimalis, thus he wants to know what is the minimum value that the product of all array elements (i.e.) can reach, if Maxim would apply no more than k operations to it. Please help him in that.

Input

The first line of the input contains three integers n, k and x $(1 \leq n, k \leq 200\ 000, 1 \leq x \leq 10^9)$ — the number of elements in the array, the maximum number of operations and the number invented by Maxim, respectively.

The second line contains n integers $a_1, a_2, ..., a_n$ () — the elements of the array found by Maxim.

Output

Print n integers $b_1, b_2, ..., b_n$ in the only line — the array elements after applying no more than k operations to the array. In particular, should stay true for every $1 \leq i \leq n$, but the product of all array elements should be **minimum possible**.

If there are multiple answers, print any of them.

Examples

input

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

output

```
5 4 3 5 -1
```

input

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

output

```
5 4 0 5 5
```

input

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

output

```
5 1 4 5 5
```

input

```
3 2 7
5 4 2
```

output

```
5 11 -5
```

E. Road to Home

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

Once Danil the student was returning home from tram stop lately by straight road of length L. The stop is located at the point $x = 0$, but the Danil's home — at the point $x = L$. Danil goes from $x = 0$ to $x = L$ with a constant speed and does not change direction of movement.

There are n street lights at the road, each of which lights some continuous segment of the road. All of the n lightened segments do not share common points.

Danil loves to sing, thus he wants to sing his favourite song over and over again during his walk. As soon as non-lightened segments of the road scare him, he sings only when he goes through the lightened segments.

Danil passes distance p while performing his favourite song once. Danil can't start another performance if the segment passed while performing is not fully lightened. Moreover, if Danil has taken a pause between two performances, he is not performing while not having passed a segment of length at least t. Formally,

1. Danil can start single performance at a point x only if every point of segment $[x, x + p]$ is lightened;
2. If Danil has finished performing at a point $x + p$, then the next performance can be started only at a point y such that $y = x + p$ or $y \geq x + p + t$ satisfying the statement under the point 1.

Blue half-circles denote performances. Please note that just after Danil has taken a pause in performing, he has not sang for a path of length of at least t

Determine how many times Danil can perform his favourite song during his walk from $x = 0$ to $x = L$.

Please note that Danil does not break a single performance, thus, started singing another time, he finishes singing when having a segment of length of p passed from the performance start point.

Input

The first line of the input contains four integers L, n, p and t ($1 \leq L \leq 10^9$, $0 \leq n \leq 100\,000$, $1 \leq p \leq 10^9$, $1 \leq t \leq 10^9$) — the length of the Danil's path, the number of street lights at the road, the distance Danil passes while doing single performance and the minimum distance of pause respectively.

The next n lines describe segments lightened by street lights. i-th of them contains two integers l_i, r_i ($0 \leq l_i < r_i \leq L$) — the endpoints of the segment lightened by i-th street light. It is guaranteed that no two segments are intersecting, nesting, or touching each other. The segments are given in the order from left to right.

Output

Print the only integer — the maximum number of performances of Danil's favourite song on the path from $x = 0$ to $x = L$.

Examples

input

```
17 2 2 6
0 9
13 17
```

output

```
5
```

input

```
12 2 2 2
0 5
6 11
```

output

```
4
```

input

```
12 2 2 4
0 5
6 11
```

output

```
3
```

Note

The first sample case is just about corresponding to the picture from the statement.