A. Exams

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

Student Valera is an undergraduate student at the University. His end of term exams are approaching and he is to pass exactly n exams. Valera is a smart guy, so he will be able to pass any exam he takes on his first try. Besides, he can take several exams on one day, and in any order.

According to the schedule, a student can take the exam for the i-th subject on the day number a_i. However, Valera has made an arrangement with each teacher and the teacher of the i-th subject allowed him to take an exam before the schedule time on day b_i ($b_i < a_i$). Thus, Valera can take an exam for the i-th subject either on day a_i, or on day b_i. All the teachers put the record of the exam in the student's record book on the day of the actual exam and write down the date of the mark as number a_i.

Valera believes that it would be rather strange if the entries in the record book did not go in the order of non-decreasing date. Therefore Valera asks you to help him. Find the minimum possible value of the day when Valera can take the final exam if he takes exams so that all the records in his record book go in the order of non-decreasing date.

Input

The first line contains a single positive integer n ($1 \le n \le 5000$) — the number of exams Valera will take.

Each of the next n lines contains two positive space-separated integers a_i and b_i ($1 \le b_i < a_i \le 10^9$) — the date of the exam in the schedule and the early date of passing the i-th exam, correspondingly.

Output

Print a single integer — the minimum possible number of the day when Valera can take the last exam if he takes all the exams so that all the records in his record book go in the order of non-decreasing date.

Examples

input

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

output

```
2
```

input

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

output

```
6
```

Note

In the first sample Valera first takes an exam in the second subject on the first day (the teacher writes down the schedule date that is 3). On the next day he takes an exam in the third subject (the teacher writes down the schedule date, 4), then he takes an exam in the first subject (the teacher writes down the mark with date 5). Thus, Valera takes the last exam on the second day and the dates will go in the non-decreasing order: 3, 4, 5.

In the second sample Valera first takes an exam in the third subject on the fourth day. Then he takes an exam in the second subject on the fifth day. After that on the sixth day Valera takes an exam in the first subject.

B. Long Jumps

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

Valery is a PE teacher at a school in Berland. Soon the students are going to take a test in long jumps, and Valery has lost his favorite ruler!

However, there is no reason for disappointment, as Valery has found another ruler, its length is l centimeters. The ruler already has n marks, with which he can make measurements. We assume that the marks are numbered from 1 to n in the order they appear from the beginning of the ruler to its end. The first point coincides with the beginning of the ruler and represents the origin. The last mark coincides with the end of the ruler, at distance l from the origin. This ruler can be repesented by an increasing sequence $a_1, a_2, \ldots, a_n$, where a_i denotes the distance of the i-th mark from the origin ($a_1 = 0$, $a_n = l$).

Valery believes that with a ruler he can measure the distance of d centimeters, if there is a pair of integers i and j ($1 \le i \le j \le n$), such that the distance between the i-th and the j-th mark is exactly equal to d (in other words, $a_j - a_i = d$).

Under the rules, the girls should be able to jump at least x centimeters, and the boys should be able to jump at least y ($x < y$) centimeters. To test the children's abilities, Valery needs a ruler to measure each of the distances x and y.

Your task is to determine what is the minimum number of additional marks you need to add on the ruler so that they can be used to measure the distances x and y. Valery can add the marks at any integer non-negative distance from the origin not exceeding the length of the ruler.

Input

The first line contains four positive space-separated integers n, l, x, y ($2 \le n \le 10^5$, $2 \le l \le 10^9$, $1 \le x < y \le l$) — the number of marks, the length of the ruler and the jump norms for girls and boys, correspondingly.

The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($0 = a_1 < a_2 < \ldots < a_n = l$), where a_i shows the distance from the i-th mark to the origin.

Output

In the first line print a single non-negative integer v — the minimum number of marks that you need to add on the ruler.

In the second line print v space-separated integers $p_1, p_2, \ldots, p_v$ ($0 \le p_i \le l$). Number p_i means that the i-th mark should be at the distance of p_i centimeters from the origin. Print the marks in any order. If there are multiple solutions, print any of them.

Examples

input
```
3 250 185 230
0 185 250
```
output
```
1
230
```

input
```
4 250 185 230
0 20 185 250
```
output
```
0
```

input
```
2 300 185 230
0 300
```
output
```
2
185 230
```

Note

In the first sample it is impossible to initially measure the distance of 230 centimeters. For that it is enough to add a 20 centimeter mark or a 230 centimeter mark.

In the second sample you already can use the ruler to measure the distances of 185 and 230 centimeters, so you don't have to add new marks.

In the third sample the ruler only contains the initial and the final marks. We will need to add two marks to be able to test the children's skills.

C. Riding in a Lift

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

Imagine that you are in a building that has exactly n floors. You can move between the floors in a lift. Let's number the floors from bottom to top with integers from 1 to n. Now you're on the floor number a. You are very bored, so you want to take the lift. Floor number b has a secret lab, the entry is forbidden. However, you already are in the mood and decide to make k consecutive trips in the lift.

Let us suppose that at the moment you are on the floor number x (initially, you were on floor a). For another trip between floors you choose some floor with number y ($y \neq x$) and the lift travels to this floor. As you cannot visit floor b with the secret lab, you decided that the distance from the current floor x to the chosen y must be strictly less than the distance from the current floor x to floor b with the secret lab. Formally, it means that the following inequation must fulfill: $|x - y| < |x - b|$. After the lift successfully transports you to floor y, you write down number y in your notepad.

Your task is to find the number of distinct number sequences that you could have written in the notebook as the result of k trips in the lift. As the sought number of trips can be rather large, find the remainder after dividing the number by 1000000007 ($10^9 + 7$).

Input

The first line of the input contains four space-separated integers n, a, b, k ($2 \le n \le 5000$, $1 \le k \le 5000$, $1 \le a, b \le n$, $a \neq b$).

Output

Print a single integer — the remainder after dividing the sought number of sequences by 1000000007 ($10^9 + 7$).

Examples

input
```
5 2 4 1
```
output
```
2
```

input
```
5 2 4 2
```
output
```
2
```

input
```
5 3 4 1
```
output
```
0
```

Note

Two sequences $p_1, p_2, \ldots, p_k$ and $q_1, q_2, \ldots, q_k$ are *distinct*, if there is such integer j ($1 \le j \le k$), that $p_j \neq q_j$.

Notes to the samples:

1. In the first sample after the first trip you are either on floor 1, or on floor 3, because $|1 - 2| < |2 - 4|$ and $|3 - 2| < |2 - 4|$.
2. In the second sample there are two possible sequences: $(1, 2)$; $(1, 3)$. You cannot choose floor 3 for the first trip because in this case no floor can be the floor for the second trip.
3. In the third sample there are no sought sequences, because you cannot choose the floor for the first trip.

D. Parcels

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

Jaroslav owns a small courier service. He has recently got and introduced a new system of processing parcels. Each parcel is a box, the box has its weight and strength. The system works as follows. It originally has an empty platform where you can put boxes by the following rules:

- If the platform is empty, then the box is put directly on the platform, otherwise it is put on the topmost box on the platform.
- The total weight of all boxes on the platform cannot exceed the strength of platform S at any time.
- The strength of any box of the platform at any time must be no less than the total weight of the boxes that stand above.

You can take only the topmost box from the platform.

The system receives n parcels, the i-th parcel arrives exactly at time in_i, its weight and strength are equal to w_i and s_i, respectively. Each parcel has a value of v_i bourles. However, to obtain this value, the system needs to give the parcel exactly at time out_i, otherwise Jaroslav will get 0 bourles for it. Thus, Jaroslav can skip any parcel and not put on the platform, formally deliver it at time in_i and not get anything for it.

Any operation in the problem is performed instantly. This means that it is possible to make several operations of receiving and delivering parcels at the same time and in any order.

Please note that the parcel that is delivered at time out_i, immediately gets outside of the system, and the following activities taking place at the same time are made without taking it into consideration.

Since the system is very complex, and there are a lot of received parcels, Jaroslav asks you to say what maximum amount of money he can get using his system.

Input

The first line of the input contains two space-separated integers n and S ($1 \le n \le 500$, $0 \le S \le 1000$). Then n lines follow, the i-th line contains five space-separated integers: in_i, out_i, w_i, s_i and v_i ($0 \le in_i < out_i < 2n$, $0 \le w_i, s_i \le 1000$, $1 \le v_i \le 10^6$). It is guaranteed that for any i and j ($i \ne j$) either $in_i \ne in_j$, or $out_i \ne out_j$.

Output

Print a single number — the maximum sum in bourles that Jaroslav can get.

Examples

input

```
3 2
0 1 1 1 1
1 2 1 1 1
0 2 1 1 1
```

output

```
3
```

input

```
5 5
0 6 1 2 1
1 2 1 1 1
1 3 1 1 1
3 6 2 1 2
4 5 1 1 1
```

output

```
5
```

Note

Note to the second sample (T is the moment in time):

- $T = 0$: The first parcel arrives, we put in on the first platform.
- $T = 1$: The second and third parcels arrive, we put the third one on the current top (i.e. first) parcel on the platform, then we put the secod one on the third one. Now the first parcel holds weight $w_2 + w_3 = 2$ and the third parcel holds $w_2 = 1$.
- $T = 2$: We deliver the second parcel and get $v_2 = 1$ bourle. Now the first parcel holds weight $w_3 = 1$, the third one holds 0.
- $T = 3$: The fourth parcel comes. First we give the third parcel and get $v_3 = 1$ bourle. Now the first parcel holds weight 0. We put the fourth parcel on it — the first one holds $w_4 = 2$.
- $T = 4$: The fifth parcel comes. We cannot put it on the top parcel of the platform as in that case the first parcel will carry weight $w_4 + w_5 = 3$, that exceed its strength $s_1 = 2$, that's unacceptable. We skip the fifth parcel and get nothing for it.

- $T = 5$: Nothing happens.
- $T = 6$: We deliver the fourth, then the first parcel and get $v_1 + v_4 = 3$ bourles for them.

Note that you could have skipped the fourth parcel and got the fifth one instead, but in this case the final sum would be 4 bourles.

E. Parking Lot

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

Petya's been bored at work and he is killing the time by watching the parking lot at the office. The parking lot looks from above like an $n \times m$ table (a cell of the table corresponds to a single parking spot). Some spots in the parking lot are taken, others are empty.

Petya watches cars riding into the parking lot one by one. After a car settles down at the parking spot, Petya amuzes himself by counting what maximum square of empty spots (i.e. a square subtable) can be seen on the parking lot if we look at it from above. Also, he takes notes of the square's size (side length) in his notebook.

You task is: given the state of the parking lot at the initial moment of time and the information about where the arriving cars park, restore what Petya wrote in his notebook. It is midday, so nobody leaves the lot.

Input

The first line contains three integers n, m and k — the sizes of the parking lot and the number of arriving cars after Petya started his watch ($1 \le n, m, k \le 2000$). Each of the following n lines contains m characters 'X' and '.', where 'X' means a taken spot and '.' means an empty spot. Each of the next k lines contains a pair of integers x_i, y_i — the number of row and column of the spot the corresponding car takes ($1 \le x_i \le n$, $1 \le y_i \le m$). It is guaranteed that this place was empty. You can assume that a car enters a parking lot only after the previous car successfully finds a spot.

Output

Print k integers — the length of the side of the maximum square of empty spots after the corresponding car has entered the parking lot.

Examples

input

```
7 8 4
........
X.....X.
........
........
.X......
........
........
1 5
6 4
3 5
4 6
```

output

```
5
4
4
3
```