Problem	LUKA	TRAJEKT	TELEPORT

Source file	`luka.pas`
	`luka.c`
	`luka.cpp`
		`trajekt.pas`
		`trajekt.c`
		`trajekt.cpp`
			`teleport.pas`
			`teleport.c`
			`teleport.cpp`

Input file	`luka.in`	`trajekt.in`	`teleport.in`
Output file	`luka.out`	`trajekt.out`	`teleport.out`
Time limit per test	5 seconds	5 seconds	5 seconds
Number of tests	10	10	10
Points per test	3	5	7
Total points	**30**	**50**	**70**
	150		

LUKA

Luka came to National Competition ten days earlier in order to get some hints by eavesdropping on the The Task Makers' conversations. He knows that The Task Makers take daily walks through the town of Cres, during which they quietly discuss about the tasks.

Curiously, the city of Cres is built in such a fashion, that it can be represented as a planar coordinate grid. The Task Makers start their walks at point **(0,0)**, and in **every turn** they move on the coordinate plain **in one of the four directions**: right (they increase their x coordinate by one), up (they increase their y coordinate by one), left or down.

Luka is positioned at point **(X,Y)** on the coordinate plain, and he can overhear bits of information **only** at times The Task Makers are located on the **same** position as Luka or one of the other **eight adjacent** positions.

Your task is to write a program that will find out the time on which Luka has heard the conversation.

Input data

The first line of the input file consists of two inegers X and Y, $-10000 \leq X,Y \leq 10000$, Luka's position.

In the next line is an integer K, $1 \leq K \leq 100{,}000$.

The following line holds K charachters that show us the route that The Task Makers were taking that day. Those charachters will be either 'I' - east (or right), 'S' - north (or up), 'Z' - west (or left) or 'J'-south (or down).

Output data

In the output file you have to write the times from the Task text. The numbers have be outputted in a **strictly ascending** order, and each one of them has to be in a **separate** line.

If Luka wasn't able to overhear the taskmasters, in the first and only line of the output file you have to write '-1'.

Examples

luka.in

```
2 2
3
III
```

luka.out

```
-1
```

luka.in

```
0 1
3
IJI
```

luka.out

```
0
1
```

luka.in

```
-1 0
8
JJZZSSIS
```

luka.out

```
0
1
5
6
7
8
```

TRAJEKT

Towns A and B are connected with a ferry line.

We know the legth of the ferry ride, the minimal boarding time (needed for unloading and loading), and the times of the ferry departures from the both towns.

You have to write a program that will calculate the **minimal** number of ferries needed **to keep the schedule.**

Note: the ferries ride **only** by the schedule, that is the ferry **cant't ride** if it is not their daeparture time.

Input data

First line of the input file consists of two integers K and L divided by a single space, $1 \leq K,L \leq 1000$, the legth of the ferry ride between the two towns and the minimal boarding time (in minutes).

The next line consists of an integer A, $1 \leq A \leq 1440$, number of departures from the town A. The next A lines consist of departure times from town A.

The next line consists of an integer B, $1 \leq B \leq 1440$, number of departures from the town B. The next B lines consist of departure times from town B.

All departure times for a town are given in chronological order and written in the HH:MM format (hours and minutes). If the minutes or the hour of the departure time are a one digit number, **they will be preceeded by a leading zero.**

The times are between 00:00 and 23:59.

Output data

The first and the only line of the output file should consist of the minimal number of ferries needed to keep the schedule.

Examples

trajekt.in

```
30 15
1
08:00
1
08:00
```

trajekt.out

```
2
```

trajekt.in

```
15 30
2
08:00
12:00
1
08:45
```

trajekt.out

```
1
```

trajekt.in

```
90 30
2
09:00
10:00
4
08:00
11:00
14:00
20:00
```

trajekt.out

```
3
```

TELEPORT

One morning Lucy and Jimmi woke up in an unknown spaceship. They looked around the ship, walked and walked, went through every room and corridor, but they haven't found anybody.

One day, Lucy said between two mouthfuls of cooked weed:

Lucy: "Hey Jimmi, did you see the two rooms with the teleport?"

Jimmi: "Teleport? What is a teleport? Lucy, lay down on the weed, it looks like it's hallucinogenic."

Lucy: "You know good and well who's hallucinogenic. The teleport is the thing that transports you between rooms so you don't need to walk."

Jimmi: "I call that the room-to-room-transporter. And you'r going on my nerves with that latin."

Lucy: "OK, teleport, transporter, what's the difference. So have you seen it?

Jimmi: "No."

Lucy: "That's what i thaught, because I had trouble finding it. Let's play a game, I'll walk between some two rooms using the shortest path, and I'll be using the teleport if it is shorter that way, and you have to find out where it is located."

Jimmi: "I think it's too hard. You know I'm not that smart."

Lucy: "Trust me, it's not too hard, I'll go a couple of times between different rooms, so it will be easier for you."

Jimmi: "OK, but if I guess it right, promise me that you'll lay down on that stinking weed"

Lucy: "OK, I'll cut it down to five meals per day."

The spaceship consists of rooms and corridors connecting them. We know for every corridor the time needed to pass it, ans we don't need time to pass through rooms. Lucy walked K times between **different** pairs of rooms (and used the teleport **when needed**) and every time wrote down how many seconds it took him.

Write a program that will find out **where** the teleports are located.

Input data

The first line of the input file consists of two integers N and M, $1 \leq N \leq 200$, $1 \leq M \leq 20000$, the number of the rooms and the number of corridors.

The next M lines consit of three integers A, B and T, divided by a single space. That means that the rooms A and B are connected by a corridor, and that we need T seconds to pass that corridor.

The next line consists of an integer K, $1 \leq K \leq 5000$.

The next K lines consist of three integers A, B and T divided by a single space. That means that that it took Lucy T seconds to get from room A to room B.

Note: the sollution is always **possible and unique**.

Output data

The first and the only line of the output file should consist of the room numbers of the two rooms where the teleport is situated. The first number sould be **smaller** than the second, and they should be divided by a single space.

Examples

teleport.in

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

teleport.out

```
1 3
```

teleport.in

```
5 4
1 2 5
2 3 3
1 4 6
4 5 5
2
3 5 8
4 3 3
```

teleport.out

```
2 4
```

teleport.in

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

teleport.out

```
1 4
```