

Problems

Problem	SPOJ	MO	PIZZA	LIFTOVI
Executable file	spoj.exe	mo.exe	pizza.exe	liftovi.exe
Source file	spoj.pas spoj.c spoj.cpp	mo.pas mo.c mo.cpp	pizza.pas pizza.c pizza.cpp	liftovi.pas liftovi.c liftovi.cpp
Input file	spoj.in	mo.in	pizza.in	liftovi.in
Output file	spoj.out	mo.out	pizza.out	liftovi.out
Time limit per test	5 seconds	5 seconds	5 seconds	5 seconds
Number of tests	10	10	10	10
Points per test	3	4	6	7
Total points	30	40	60	70
	200			

SPOJ

Ivica is worried about amount of money he spend for phone bill for surfing on Internet.

One minute of surfing in the period **between 7:00 and 19:00** costs **10** cents, and in period **between 19:00 and 7:00** costs **5** cents (the price is the same for all days in the months).

Start and end of one connection to Internet is always rounded **to minute (not seconds)**, and one connection lasts **at most 60 minutes**.

You are given a list of connections, write a program which will calculate the **total amount** of money spent on Internet.

Input data

In the first line of input file there is integer N , number of connections, $1 \leq N \leq 100$.

In each of next N lines there is description of one connection, in the following format:

HH:MM DD

with HH:MM denotes hour and minute of start of connection, and DD denotes for duration of connection (in minutes, maximum 60). Between MM and DD there is one space character.

If hour or minute of start of connection or duration is one-digit number, then there is a **leading zero** in front of that number.

Times are denoted from 00:00 till 23:59.

Output data

You have to write total amount of money (in cents) in only line in output file.

Examples

spoj.in

```
2
11:02 11
15:30 01
```

spoj.out

```
120
```

spoj.in

```
3
20:05 12
06:45 30
13:08 15
```

spoj.out

```
435
```

spoj.in

```
5
00:00 05
06:47 35
11:30 18
18:33 60
23:59 22
```

spoj.out

```
1035
```

MO

Mirko and Slavko are playing MO (mini-go). MO is similar to ancient chinese game GO, but is simpler and it plays on (**one-dimensional**) **table** which consists of certain number of squares in the sequence.

Mirko is playing with **white** coins, and he plays **first**. Slavko is playing with **black** coins, and he plays **second**.

In the beginning of the game, all the squares on the table are **empty**. The players are playing **one by one** and in each move they put **one** of their coins on any **empty** square on the table.

If there are **consequent sequence** of opponent's coins **between just putted** coin and **previously putted coin with same color**, then that sequence of opponent's coins are **removed** from the table.

Squares on the table are denoted with numbers from 1 till P, from left to right.

Write a program which will calculate number of **white** and number of **black** coins on the table on the **end of the game**.

Input data

In the first line there are two integers P and N, separated with one space character, number of squares on the table and total number of moves of both players, $1 \leq P \leq 100$, $1 \leq N \leq 1000$.

In each of the next N lines, there are moves of Mirko and Slavko, from the beginning to the end of the game.

Output data

In the only line of the output file you have to write number of white and number of black coins in the end of the game. That numbers must be separated with one space character.

Examples

mo.in

4 4
2
3
4
3

mo.out

2 1

mo.in

5 6
1
4
5
2
3
2

mo.out

3 1

mo.in

6 8
1
2
5
3
4
6
2
3

mo.out

2 2

PIZZA

Our friend Picko is very reach and he wants to open lots of restaurants with delivery. The main food will be, of course, pizza.

He has certain number of potential locations for the restaurants, and he knows the locations of the solitaires with lots of people which will often be his customers.

Delivery of each restaurant will cover all the solitaires in given radius. Picko can open only limited number of restaurants, and he wants that restaurants on the locations which will cover maximal number of people in solitaires.

Write a program that will calculate **maximal** number of people which we can cover with delivery.

Input data

In the first line of the input file there are two integers K and R , separated with space, number of restaurants and radius of delivery, $1 \leq K \leq 10$, $1 \leq R \leq 500$.

In the second line there is integer M , number of locations, $K \leq M \leq 20$.

In each of the next M lines there are two integer X and Y , separated with space, coordinates of each location, $-1000 \leq X, Y \leq 1000$.

In the next line there is integer N , number of solitaires, $1 \leq N \leq 100$.

In each of the next N lines there are three integers X , Y and S , separated with space, X and Y are coordinates of each solitaire, and S is number of people in that solitaire, $-1000 \leq X, Y \leq 1000$, $1 \leq S \leq 100$. We consider that solitaire is in radius of some restaurant if distance between them is **less or equal to R** .

There are no two locations of restaurants on the same place.

Output data

In only line of the output file we have to write maximal number from the text above.

Examples

pizza.in

```
2 2
3
1 0
4 0
7 0
4
0 0 1
3 0 7
5 0 9
8 0 1
```

pizza.out

```
18
```

pizza.in

```
2 2
3
-2 0
0 1
3 0
8
-3 1 1
-3 0 1
-3 -1 1
-2 -1 1
0 0 3
0 2 1
2 1 3
4 0 2
```

pizza.out

```
12
```

pizza.in

```
3 3
5
0 0
1 6
2 3
6 6
7 2
8
0 1 2
0 5 3
0 6 1
1 0 1
3 2 3
3 6 2
6 2 4
8 6 3
```

pizza.out

```
17
```

LIFTOVI

Solitaire has N elevators. Each elevator are connecting **exactly two** floors and it **does not** stop on the floors between that two floors. The speed of all the elevators are the **same, 5 seconds to pass one floor**.

On the beginning, each elevator is in its **lower** position and they are starting cruising to the upper floor. After some elevator come to its upper position, it **immediatly** starts to go back to its lower position, and so on...

Mirko is on the **first (the lowest)** floor and he wants **as quick as possible** come to the **top of the solitaire**. He can change elevators **only** on the floors that are common to both elevators, and if the other elevator is **in that moment** on that floor, that change **does not** take any time.

Write a program that will calculate **minimal** time in which Mirko can get to the top of the solitaire.

Input data

In the first line of the input file there are two integers K and N , separated with space, number of floors in solitaire and number of elevators, $2 \leq K \leq 1000$, $1 \leq N \leq 50000$.

In each of the next N lines there are description of one elevator, two integers A and B , separated with space, $1 \leq A < B \leq K$, means that elevator is travelling between floors A and B .

There are no two different elevators that travels between **same** floors.

Note: input data will guarantee that solution will always exists.

Output data

In the only line of output file write minimal time (in seconds) from the text above.

Examples

```
liftovi.in
10 4
1 5
5 10
5 7
7 10

liftovi.out
45
```

```
liftovi.in
10 3
1 5
3 5
3 10

liftovi.out
105
```

```
liftovi.in
20 5
1 7
7 20
4 7
4 10
10 20

liftovi.out
150
```