

Problems

Problem	BUHA	LIGNJA	ONE
Source file	buha.pas buha.c buha.cpp	lignja.pas lignja.c lignja.cpp	one.pas one.c one.cpp
Input file	buha.in	lignja.in	one.in
Output file	buha.out	lignja.out	one.out
Time limit per test	10 seconds	10 seconds	10 seconds
Number of tests	10	10	10
Points per test	5	7	8
Total points	50	70	80
	200		

BUHA

Boško and Suško are playing an interesting game on a board made of rectangular fields arranged in A rows and B columns.

When the game starts, Suško puts its virtual pillbox on **one** field the board. Then Boško selects fields on which he will throw his virtual bombs and Suško replies for each bomb if his bunker is in a range.

Range of bomb with diameter P that is thrown on field (R,S) is a rectangle with sides parallel to sides of the board, with field (R,S) in its center and length of each side is P .

Write a program that will help Boško to find out on **how many** fields could have been placed Suško's pillbox after some bombs have been thrown.

Input data

First line of input file contains three integers: A , B and K , $1 \leq A,B,K \leq 100$. A represents number of rows, B number of columns and K number of thrown bombs.

Each of the next K lines contains integers R , S , P and T , describing a bomb thrown at the field in R th row and S^{th} column with diameter P , $1 \leq P \leq 99$, P is odd. T equals 1 if the pillbox is in a range if that bomb, otherwise it is 0.

Output data

Write to the output file the number of fields on which could have been placed bunker.

Examples

buha.in

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

buha.out

```
4
```

buha.in

```
5 5 3
3 3 3 1
3 4 1 0
3 4 3 1
```

buha.out

```
5
```

buha.in

```
5 4 1
2 1 5 0
```

buha.out

```
8
```

LIGNJA

Every day before Mirko goes to work, he connects to the Internet and reads mail from his boss that contains a list of Mirko's jobs for that day. Each job is defined by its **starting time** and **duration**.

Mirko's work shift lasts N minutes. When Mirko arrives to work, he starts processing his jobs. If there are **more** jobs he has to start working on at the **same** time, he can **choose** which one he will process and other jobs will be processed by his co-workers. When he finishes one job, he reads newspapers until the next job starts. If Mirko is busy processing one job when another job starts, he will not process that other job not even after he finishes the current one.

When more jobs are starting at the same time by choosing which job to process Mirko can increase the amount of time he will be free to read newspapers.

Write a program that will help Mirko decide which jobs to process in order to maximize his free time.

Input data

First line of the input file contains two integers: N and K , $1 \leq N \leq 10000$, $1 \leq K \leq 10000$. N represents how many minutes lasts Mirko's work shift. K represents number of jobs.

Each of the next K lines contains data about one job: integers P and T , meaning that job is starting in P th minute and its duration is T minutes. $1 \leq P \leq N$, $1 \leq P+T-1 \leq N$.

Output data

Write to the output file the maximum number of minutes Mirko can spend reading newspapers.

Examples

lignja.in

```
10 3
1 2
4 3
5 1
```

lignja.out

```
5
```

lignja.in

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

lignja.out

```
5
```

lignja.in

```
15 6
1 2
1 6
4 11
8 5
8 1
11 5
```

lignja.out

```
4
```

ONE

The city consists of intersections and streets that connect them.

Heavy snow covered the city so the mayor Milan gave to the winter-service a list of streets that have to be cleaned of snow. These streets are chosen such that the number of streets is **as small as possible** but still every two intersections to be **connected** i.e. between every two intersections there will be **exactly one** path. The winter service consists of **one** snow plover and one driver Mirko and his starting position is on one of the intersections.

The snow plover burns **one liter of fuel** per meter (even if it is driving through a street that has already been cleared of snow) and it has to clean **all** streets from the list in such order so the total fuel spent is **minimal**. When all the streets are cleared of snow, the snow plover is parked on the last intersection it visited.

Write a program that calculates the total amount of fuel that the snow plover will spend.

Input data

The first line of the input file contains two integers: N and S , $1 \leq N \leq 100000$, $1 \leq S \leq N$. N is the total number of intersections; S is ordinal number of the snow plover starting intersection. Intersections are marked with numbers $1..N$.

Each of the next $N-1$ lines contains three integers: A , B and C , meaning that intersections A and B are directly connected by a street and that street's length is C meters, $1 \leq C \leq 1000$.

Output data

Write to the output file the minimal amount of fuel needed to clean all streets.

Examples

one.in

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

one.out

```
7
```

one.in

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

one.out

```
5
```

one.in

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

one.out

```
14
```