TASK	FILIP	SLATKIS	SORT	RAZGOVOR	PATULJCI	PLANETE
source code	filip.pas filip.c filip.cpp	slatkisi.pas slatkisi.c slatkisi.cpp	sort.pas sort.c sort.cpp	razgovori.pas razgovori.c razgovori.cpp	patuljci.pas patuljci.c patuljci.cpp	planete.pas planete.c planete.cpp
input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 s	1 s	1 s	1 s	1 s	1 s
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
point value	**30**	**50**	**70**	**100**	**120**	**130**
	500					

Mirko has a younger brother, Filip, who just started going to school and is having trouble with numbers. To help him get the hang for number scale, his teacher writes two three digit numbers. She asks Filip to compare those numbers, but instead of interpreting them with the leftmost most significant digit, he needs to interpret them the other way around, with the most significant digit being the rightmost one. He than has to tell the teacher the larger of the two numbers.

Write a program that will check Filips answers.

INPUT

The first and only line of input contains two three digit numbers, **A** and **B**. **A** and **B** will not be equal and will not contain any zeroes.

OUTPUT

First and only line of output should contain the larger of the numbers in the input, compared as described in the task. **The number should be written reversed, to display to Filip how he should read it.**

SAMPLE TESTS

input	**input**	**input**
734 893	221 231	839 237
output	**output**	**output**
437	132	938

Mirko buys a lot of candy in the candy shop. He cannot always pay the exact ammount so the shopkeeper and he have an agreement. He tells the shopkeeper the smallest bill he has, and she rounds his ammount to the nearest number he can pay. For example, if the smallest bill Mirko has is a hundred bill, and he wants to buy 150 Kunas of candy, the shopkeeper rounds his ammount to 200 Kunas. If he wants to buy 149 Kunas of candy, the shopkeeper rounds his ammount to 100 Kunas.

Lately, Mirko suspects the shoopkeeper is trying to cheat him. He asked you to help him. Write a program that will help him.

His mother only gives Mirko 1, 10, 100, 1 000, ... , 1 000 000 000 Kuna bills. He never has bills that are not powers of 10. **The bills he does have, he has in large ammounts.**

INPUT

The first line contains an integer, **C** (0 ≤ **C** ≤ 1 000 000 000), the price of candy Mirko is going to buy.
The second line contains an integer, **K** (0 ≤ **K** ≤ 9), number of zeros on the **smallest** bill Mirko has.

OUTPUT

The first and only line of output should contain one integer, **C** rounded to the nearest amount Mirko can pay.

SAMPLE TESTS

input 184 1 **output** 180	**input** 123450995 1 **output** 123451000	**input** 182 2 **output** 200

Author: Luka Kalinovčić

Mirko is a great code breaker. He knows any cipher in the world can be broken by frequency analysis. He has completely the wrong idea what frequency analysis is, however.

He intercepted an enemy message. The message consists of **N** numbers, smaller than or equal to **C** Mirko belives freqency analysis consists of sorting this sequence so that more frequent numbers appear before less frequent ones.

Formally, the sequence must be sorted so that given any two numbers **X** and **Y** , **X** appears before **Y** if the number of times **X** appears in the original sequence is larger than the number of time **Y** does. If the number of appearances is equal, the number whoes **value** appears sooner in the input should appear sooner in the sorted sequence.

Help Mirko by creating a "frequency sorter".

INPUT

First line of input contains two integers, **N** (1 ≤ **N** ≤ 1 000), length of message, and **C** (1 ≤ **C** ≤ 1 000 000 000), the number from task description.

Next line contains **N** integers smaller than or equal to **C**, message itself.

OUTPUT

First and only line of output should contain **N** numbers, the sorted sequence.

SAMPLE TESTS

input	input	input
5 2 1 2 1 2	9 1 3 3 3 2 2 2 1 1	9 11 33 11 77 54 11 25 25 33
output	**output**	**output**
2 2 2 1 1	1 1 1 3 3 3 2 2 2	11 11 11 33 33 25 25 77 54

Mirkos village has only one long street stretching from east to west with **M** houses. Each house has a unique house number, starting with 1 and ending with **M**.

Recent storm took out most phone lines so the mayor financed construction of a new one. Mirko is interested in the popularity of this new phone network, so he infiltrated its construction and placed special detectors on some points.

Detector detects **any** phone call made between two houses, as long as one of them is **eastward** and the other **westward** from the point the detector is installed.

At the end of the first month, Mirko removed all detectors and now wonders what is the **smallest number** of phone calls that could have been made during that month.

INPUT

The first line of input contains two integers **N** ($1 \leq N \leq 100\,000$), number of detectors, and **M** ($N < M \leq 1\,000\,000\,000$), number of houses in the village.

Next **N** lines contains two numbers each: **P**i ($1 \leq P_i < M$), and C_i ($1 \leq C_i \leq 1\,000\,000\,000$), the position and total number of phone calls detected by detector numbered **i**. We say that a detector is on position P_i if and only if he is between houses numbered P_i and P_i+1.

There will never be more than one detector on the same position.

OUTPUT

Output a single integer, the minimal number of phone calls made.

GRADING

In test cases worth 50% points **N** and **C** will be smaller than 1 000.

SAMPLE TESTS

<table>
<tr>
<td>input

3 4
3 1
2 2
1 1

output

2</td>
<td>input

2 3
1 23
2 17

output

23</td>
<td>input

3 9
7 2
8 3
3 4

output

5</td>
</tr>
</table>

Snow White and the **N** dwarfs live in the forest. While the dwarfs mine away Snow White hangs around social networks.

Each morning the dwarfs form a long line and go whistling away to the mine. Snow White runs around them and snaps pictures to upload onto her favorite social network.

When dwarfs enter the mine, Snow White goes back to their house and goes through the pictures, selecting pretty ones. Each dwarf has a colored cap, and there are **C** different colors. A picture is pretty if more than half caps on it are of the same color. In other words, if there are **K** dwarfs on the picture, it is pretty if strictly more than **K / 2** dwarfs have same colored caps.

Write a program that will check for a set of **M** pictures if they are pretty, and what color is dominating if they are.

INPUT

First line contains two integers **N** and **C** (3 ≤ **N** ≤ 300 000, 1 ≤ **C** ≤ 10 000) number of dwarfs and number of colors.

Second line contains **N** integers between 1 and **C** (inclusive), colors of dwarves hats, ordered the way they formed the line that morning.

Third line contains **M** (1 ≤ **M** ≤ 10 000), number of pictures.

Next **M** lines contain two integers **A** and **B** (1 ≤ **A** ≤ **B** ≤ **N**). Each line describes one picture. On it there are all dwarves starting from **A**-th all the way to the **B**-th.

OUTPUT

Output **M** lines. For each picture output “no” if Snow White doesn't think the picture is pretty, and “yes **X**”, where **X** is the color dominating on the picture, if she does.

GRADING

In test cases worth 30% points, **M** will be smaller than 10.

In test cases worth aditional 30% points, **C** will be smaller than 10.

SAMPLE TESTS

input

```
10 3
1 2 1 2 1 2 3 2 3 3
8
1 2
1 3
1 4
1 5
2 5
2 6
6 9
7 10
```

output

```
no
yes 1
no
yes 1
no
yes 2
no
yes 3
```

Author: Goran Žužić, Luka Kalinovčić

European Space Agency has **N** telescopes observing distant planets. The classified **M** different types of events. They noticed that every day on each planet there is **exactly one** type of event. They measure events in days because they discovered each event lasts at least one day, and at most 365 days. Events last only complete number of days, they never last fractional part of day. Each event of the same type lasts exactly the same number of days every time it occurs, **regardless of the planet it occurs on**.

After many years of observation they want to analyze data. The telescopes logged all observations, but unfortunately they forgot to write down the year the observations took place! Each telescope entry consists of start date, end date and number of each type of events observed. Given all observations on all planets, calculate the duration of each type of event. Note that the telescopes are super precise and they start working exactly when the event starts and end when an event ends, not necessarily the same one.

Note: for this task, each year has exactely 365 days.

INPUT

First line contains integers **N** and **M** ($1 \le$ **N**, **M** ≤ 200) number of telescopes and number of event types. Next **N** lines contain **M+4** numbers each:

„D_1D_1 M_1M_1 D_2D_2 M_2M_2 F_1 F_2 ... F_M“

Where D_1D_1 M_1M_1 ($01 \le DD \le 31$, $01 \le MM \le 12$) are start date, D_2D_2 M_2M_2 end date, and F_i ($0 \le F_i \le 200$) number of times event type **I** was observed.

OUTPUT

One line containing **M** numbers, durations of event types. If more than one solution exists, output any one. If no solution exists, output "-1".

Author: Goran Žužić, Luka Kalinovčić

SAMPLE TESTS

<table>
<tr>
<td>input

1 1
26 02 03 03 1

output

5</td>
<td>input

1 1
26 02 03 03 2

output

185</td>
<td>input

3 3
22 03 01 10 9 10 10
05 05 16 12 1 7 10
20 06 15 01 4 9 10

output

104
204
125</td>
</tr>
</table>

First sample description:

Telescope observed one type 1 event between 26. February and 3. March. The only possible explanation is that type 1 events last 5 days.