The contest is in progress. It ends about 9 hours from now.

IEEE.tv Problem

In this problem we would like your assistance so as to put new floor in IEEE TV main floor. But let's take everything from the beginning.

IEEE TV wants to put new tiles to your local IEEE student branch to prepare the environment for them to shoot a documentory video about your Student branch activities. To save some money, they would like to minimize the cost of the tiles needed. Peter is our ideal guy for the job. Whenever he starts a new room, he starts from the most top left point and puts tiles in rows. Whenever a row of tiles ends, he continues with the next row. However, when it comes to put a whole tile that doesn't fit, then he tries to minimize the use of tiles in his own way. Firstly, he tries to find the tile with minimum width from the used ones that is equal or bigger in width than the space needed. Then he cuts that tile to that width, and then cuts again to the height he needs, creating -probably- two other smaller tiles apart from the one he puts on the floor. If there more than one tiles with the optimal width he can find, he uses that one that has the minimum height that can pay the bill. Of course, if he can't find a tile that can do the job, then he gets a new one and goes to cut it in the predetermined procedure. However, he never rotates the tiles.

Your job is to help us find how many tiles Peter needs, and foresee what unused ones will be thrown away. You will be given a list of tiles (Auto-increment number, width, height) and a list of rooms (one Latin character as ID, tile id, width of room, height of room) and you are requested to A) print how rooms will be constructed (layout), B) How many tiles will be used and what unused ones will be thrown away, per tile given, or '-' if there are no tiles thrown away.

Input

The first line will have two numbers.

- Number of tiles (number-of-tiles <= 20)

- Number of rooms (number-of-rooms <= 24)

Following to the first line, there are few lines equal to the number of tiles. Each line consists of three numbers separated by a space.

- An auto-increment number (tile-id)
- Width of the tile
- Height of the tile

Afterwards, there are few lines equal to the number of rooms. Each line consists of the following:

- Room id (a single Latin character)
- Id of tile used for this room (tile-id)
- Width of room
- Height of room

Output

For each room, first print the id of the room (single Latin character) in one line, and then the layout of that room using the character X (ASCII 88) and O (ASCII 79) to simulate the positioning of the tiles. Each line will be ended by the character | (ASCII 124). Rooms should be printed in the order given in the input data.

Then, for each tile, you should print its id (tile-id) in a single line. Then, how many new tiles will be used in general, also in a single line. Finally, a list of the remaining used tiles in descending order of width and then of height (for these that the width is the same). If there no remaining used tiles, your program should print a dash ('-') in a single line. Tiles should be printed in the order given in the input data. Please, pay attention to the following example to understand the structure of input and output data.

Sample input 1:

```
2 2
1 3 3
2 2 4
A 1 12 13
B 2 17 5
```

Sample output 1:

```
A
XXXOOOXXXOOO|
XXXOOOXXXOOO|
XXXOOOXXXOOO|
OOOXXXOOOXXX|
OOOXXXOOOXXX|
OOOXXXOOOXXX|
XXXOOOXXXOOO|
XXXOOOXXXOOO|
XXXOOOXXXOOO|
OOOXXXOOOXXX|
OOOXXXOOOXXX|
OOOXXXOOOXXX|
XXXOOOXXXOOO|
B
XXOOXXOOXXOOXXOOX|
XXOOXXOOXXOOXXOOX|
XXOOXXOOXXOOXXOOX|
XXOOXXOOXXOOXXOOX|
OOXXOOXXOOXXOOXXO|
1
18
3 2
2
11
1 3
```

Problem Author: IEEE

Suggest Edits

Emacs | Normal | Vim

Select Language: C#

save code

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 class Solution {
5     static void Main(String[] args) {
6         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class
  should be named Solution */
7     }
8 }
```

Line: 1 Col: 1 Count: 246

☐ **Use a custom test case**

Upload Code as File

Compile & Test | Submit Code