Blue

Time Limit: 5 seconds

As an Arts student at heart, one of your favourite pastimes is strolling around the local art gallery and contemplating with awe the creations of the imagination. One day you imagine how nice it would be to tile the entrance of your mansion to look like one of these artworks, and so you pull out your digital camera and photograph one of your favourite paintings. Unfortunately, being the arty person you are, you accidentally left your blue filter on your camera, and so your photo has turned out to be coloured entirely in various shades of blue.

This doesn't faze you though, as you're quite comfortable with the idea of a blue-tiled entrance. After taking a look around your local tile shop, you find a number of blue-shaded tiles that you can use. All of the tiles are square, with sizes 1m × 1m, 2m × 2m, 3m × 3m or 4m × 4m. Each tile is coloured in a single shade of blue, which you can measure using an integer ranging from 0 (very dark blue) to 255 (very light blue). Each pixel of your photograph can also be measured from 0 to 255 on this same scale.

Your task is to lay out a collection of these tiles in the entrance to your mansion, where a 1m × 1m portion of tile corresponds to a single pixel of your photograph. The *error* for each pixel is the absolute difference between the shade of the pixel and the shade of the corresponding piece of tile. The *total error* for the entire picture is the sum of the individual errors for every pixel. You must make your entrance resemble the photograph as closely as possible, i.e., you must make the total error as small as possible.

Tiles may not overlap each other, and they may not extend beyond the boundary of the picture. You may not leave any holes (i.e., pixels of your photograph for which there is no corresponding piece of tile). Although the shop only offers a few different types of tile, you may purchase as many of each type of tile as you like.

Input

Your program must read from standard input. The first line will be a single integer n ($1 \leq n \leq 20$), denoting the number of different types of tiles that the shop sells. These types are numbered $1, 2, \ldots, n$.

Following this will be n lines each describing a single type of tile that the shop sells. Each of these lines will have the form s k, where s represents the side length (in metres) of the square tile and k is an integer describing the shade of blue that the tile is coloured. You are guaranteed that $1 \leq s \leq 4$ and that $0 \leq k \leq 255$. There will always be at least one tile of dimension 1m × 1m in the input.

After this will be a line consisting of two integers h w, where h and w are the height and width of your photograph measured in pixels ($1 \leq h, w \leq 200$). Following this will be h lines of w integers each (separated by spaces), describing the shade of each pixel of your photograph. Each of these integers will be between 0 and 255 inclusive.

Output

Your program must write the best solution it can find to standard output. Your output should contain one line for each tile that you place. Each of these lines should have the form r c t, where r and c are the row and column of the top-left corner of the tile ($1 \leq r \leq h$ and $1 \leq c \leq w$), and t is an integer describing which type of tile you are using ($1 \leq t \leq n$). You may output these tiles in any order you like.

After the tiles have been written, your program must write one additional line of output. This line must contain a single integer representing the total error for your solution.

Sample Input

```
3
1 10
2 15
1 20
3 4
16 15 10 25
14 15 14 30
10 10 30 11
```

Sample Output

```
1 1 2
3 1 1
3 2 1
1 3 1
1 4 3
2 3 2
42
```

The following three diagrams demonstrate the sample data above. The diagram on the left is the photograph you are required to replicate with as little error as possible. The central diagram is one possible tiling of this, using only the tiles given in the sample input. The diagram on the right shows the resulting error for each pixel of the photograph.

16	15	10	25
14	15	14	30
10	10	30	11

15	15	10	20
15	15	15	15
10	10	15	15

1	0	0	5
1	0	1	15
0	0	15	4

The sum of all errors for each pixel is $1 + 0 + 0 + 5 + 1 + 0 + 1 + 15 + 0 + 0 + 15 + 4 = 42$, hence the total error for this tiling is 42.

Scoring

There is no particular "best solution" that you are required to achieve. Instead, your score will be determined relative to the other contestants whom you are competing against (as well as the judges' solution). For each test case, the contestant who achieves the least total error is identified. Your score for this test case will then be:

- 100% if your program finds a solution with this same least total error;
- 10% if your program simply matches each pixel of the photograph with the 1m × 1m tile closest to that pixel's shade (ignoring all tiles of size 2m × 2m or above);
- 0% if your program generates an incorrect solution;
- otherwise determined by a linear scale according to your total error, with the 100% and 10% marks on the scale corresponding to the solutions described above. If your output contains a correct tiling, no matter how large your total error is, you will be guaranteed at least 5%.

For example, for the above sample data, the total error achieved by matching each pixel with the closest 1m × 1m tile is 48. If the best solution found by any contestant (or the judges) has a total error of 32, then the scoring scale for a *correct* solution would be as follows:

Total error	32	34	36	38	40	42	44	46	48	50	52	54
Score	100%	89%	78%	66%	55%	44%	33%	21%	10%	5%	5%	5%