Round I. Parachuting

Winney the Pooh and his friends followed closely the first edition of CodeIT. They really wanted to take part in it. However, they were hampered by the slow Wi-Fi connection in the forest. They felt sorry for the missed opportunity to win the grand prize in the contest and thus decided to organize a real adventure themselves. At first they tried to get a ride on the local “Maserati”, but some other group of animals had already rented the car. So they moved on to plan B: parachute jumping.

However, before every adventure, one must think about safety first. Winney already determined the place for landing. It is a field, which can be represented as an N by M matrix. Each cell from this matrix determines how safe the corresponding part of the field is. Winney also has a list with all P animals, which want to jump. In addition, for each animal are known:

1) Its shape during the landing. In other words, which parts of the animal have direct contact with the ground and therefore can be injured. Each shape will consist of a single connected component without any unconnected parts inside of it. Two cells are connected if they have a common side. The animals cannot rotate during the flight. Furthermore, their landing shape is fixed and cannot be rotated.
2) Its impact coefficient. After every landing the place where the animal has landed becomes more and more dangerous. We will not explain the sophisticated physical laws related with the impact as they result in a simple integer division: the safety value of each cell the animal has landed on is divided by its impact coefficient.
3) Its safety threshold. During every landing it is very important that there would not be any parts from the animal which land on cells with safety value below its safety threshold.

Let the safety of a jump be defined as the sum of the safety values of all the cells which are affected by the jumping animal. Do not forget that after the jump those cells will decrease as much times as the coefficient of the corresponding animal. For more details, see the sample test cases.

Help Winney the Pooh by writing a program, which will make such a jumping schedule that the total safety from all the jumps performed will be as much as possible. It is not required that all the listed animals should actually make a jump.

Input:

The first line of the input file **parash.in** consists of three integers N, M and P. Each of the next N lines contains M space separated integers – the safety values defining the field. The next lines contain information about every individual animal. The i^{-th} animal is described by four integers – r_i, c_i, k_i and t_i, where r_i and c_i are the dimensions of the table which defines the landing shape of the animal (or simply shape table), k_i is its impact coefficient and t_i is its safety threshold. Then follows the description of the animal’s landing shape - r_i lines with c_i characters each. Each symbol is either 0, if the corresponding cell does not contain a part of the animal or 1, otherwise.

Output:

On the first line of the output file **parash.out** your program must print a single integer B denoting the number of animals which will jump according to the schedule. On the next B lines it should print 3 integers given in that order – the index of the animal which is going to jump next as well as the row and column of the cell, coinciding with the upper left corner of the animal's shape table. The animals are indexed from 1 to P according to the order in which they appear in the input file. Both the rows and columns are indexed from 1.

Limitations:

2 ≤ **N, M** ≤ 50
1 ≤ **P** ≤ 100
1 ≤ $\mathbf{r_i}$ ≤ min(N, 10)
1 ≤ $\mathbf{c_i}$ ≤ min(M, 10)
2 ≤ $\mathbf{k_i}$ ≤ 1000
1 ≤ $\mathbf{t_i}$ ≤ 1000
1 ≤ Safety value of each cell from the field ≤ 100 000

The shape table of each animal would contain neither rows nor columns consisting entirely of zeros.

Scoring:

The scoring formula is $\left(\frac{\mathbf{yours\ +1}}{\mathbf{max+1}}\right)^{\mathbf{2}} \times \frac{\mathbf{100}}{\mathbf{T}}$, where **max** is the maximal total safety, obtained by a participant, **yours** is the total safety obtained by your program and **T** is the number of test cases. You will receive 0 points for a given test case if the schedule found by your program lands an animal outside the field or if there is an animal landing on a cell bellow its safety threshold.

The time limit for each test case is 2 seconds.

Example:

parash.in:	parash.out:	Explanation:

5 5 2
1 9 4 4 7
1 6 7 7 5
6 2 1 6 6
6 7 8 2 5
1 4 2 2 3
3 3 3 2
010
110
111
3 2 2 4
01
11
01
	2
	1 2 1
	2 1 4
		This is one possible solution. The first animal will jump on the cells with upper left corner (2, 1). The safety of the jump is 6+2+6+6+7+8 = 35. After the jump all affected by the animal cells will decrease 3 times.
		
		The second animal will jump on the cells with upper left corner (1, 4), with safety 7+7+5+6 = 25. After the jump the affected cells decrease 2 times.
		
		The total safety from the jumps is 60.

Testing details:

The following table provides you with additional information about the test cases which will be used for determining the final results of the round:

% of the official test cases	Type 1 (20 %)	Type 2 (30 %)	Type 3 (50 %)
N ≤	10	25	50
M ≤	10	25	50
P ≤	20	50	100
k_i ≤	10	100	1000
t_i ≤	10	100	1000
Max safety value of a cell from the field	1000	10 000	100 000

Generator:

The contestants will have access to the generator which will be used for making the final tests. You will find it on the contest website.

The generator has three modes of execution. Each mode can be run under the command line (*command prompt* under *Windows*; *terminal* or *konsole* under *Linux*):

1) First mode (no additional parameters):

Operating system	Linux	Windows
Command line	./parash.gen	parash.gen

By default, SEED = 1 (see the second mode for more details) and the limitations used are the same as those listed in the “Limitations” section.

2) Second mode (with a given SEED):

Operating system	Linux	Windows
Command line	./parash.gen 257	parash.gen 257

By using this mode, you can select the random seed (SEED) with which the test case will be generated. All numbers in the test will be generated according to this number. In the example above, SEED = 257. Note that the output of the generator when using constant seed will be the same on every execution.

By default, the limitations used are the same as those listed in the “Limitations” section.

3) Third mode (with a given SEED and a file, defining the constraints in the task):

Operating system	Linux	Windows
Command line	./parash.gen 257 constr.txt	parash.gen 257 constr.txt

Again, just like in the second mode, you can select the random seed, generating the test case. The difference between the two modes is that in this one you can specify the limitations used during generation. For that purpose, you should make a file containing **exactly** 6 integers – the maximal allowed values for N, M, P, k_i, t_i and for safety of a cell from the field, respectively. The minimal values for those numbers are the same as the ones listed in the “Limitations” section. In the example above, those limitations are specified the “constr.txt” file, which is in the same directory as the generator. If the file you made is in a different directory, you should specify the complete path to it.