

## (Adv.) Competitive Programming

Submit until 05.07.2019 13:30, via the [judge interface](#)



### Problem: bachelorprojects (2 second timelimit)

Its July again and the current fourth semester students are choosing their bachelorprojects. But this year there are a few things new. The main difference is, that the algorithm is not longer Prof. Döllner himself, no this time there should be a algorithm, which directly finds the best possible solution. Secondly each student has to select his five favourite projects and give them the priorities 1 to 5 (where the project with the priority 1 is the project with the highest priority).

After the deadline for the submissions, Prof. Döllner has the priorities of all students and he wants to compute a assignment, so that the sum over all priorities is as small as possible. Since he knows, that you have some great knowledge about algorithms, please help him finding a good distribution.

**Input** The input starts with one line, containing the numbers  $n$  ( $1 \leq n \leq 200$ ),  $p$  ( $1 \leq p \leq 25$ ),  $m$  ( $1 \leq m \leq 15$ ) and  $k$  ( $1 \leq k \leq 20$ ).  $n$  is the number of students,  $p$  is the total number of projects,  $m$  is the number of priorities, each student gave to Prof. Döllner ( $k \leq p$ ) and  $k$  is the amount of people in each project. You can assume, that each project has the same amount of students and that  $n = p * k$  is true.

Each of the following  $n$  lines contains the priorities of one student. Every line starts with a number  $i$ , the id of the student (starting at 0). Afterwards  $m$  numbers are following. The first of the  $m$  numbers  $m_0$  is the id of the project (which starts with 0) with the priority number 1.  $m_1$  is id of the project with the second highest priority and  $m_{m-1}$  is the project with the least priority. ( $\forall i < m, \forall j < m : i \neq j \Leftrightarrow m_i \neq m_j$ ) All other projects have automatically the priority  $2m$ .

**Output** Please return an assignment where the sum over all priorities is as small as possible. Note, that it is possible, that a student gets a project, which he hasn't selected in his priority-list.

The first line of the output should contain the sum of priorities, which should be as small as possible. Afterwards print the assignment in  $n$  lines (sorted by the id of the student). Each line should contain two numbers  $i$  and  $j$ , where  $i$  is the id of the student and  $j$  is the id project where he has been assigned to. Every valid assignment where the sum over all priorities is as small as possible, will be accepted by the judge.

**Sample input**

```
6 3 2 2
0 2 0
1 0 2
2 0 1
3 0 1
4 2 1
5 0 2
```

**Sample output**

```
8
0 2
1 0
2 1
3 1
4 2
5 0
```