1006 Fixed Partition Memory Management

A technique used in early multiprogramming operating systems involved partitioning the available primary memory into a number of regions with each region having a fixed size, different regions potentially having different sizes. The sum of the sizes of all regions equals the size of the primary memory.

Given a set of programs, it was the task of the operating system to assign the programs to different memory regions, so that they could be executed concurrently. This was made difficult due to the fact that the execution time of a program might depend on the amount of memory available to it. Every program has a minimum space requirement, but if it is assigned to a larger memory region its execution time might increase or decrease.

In this program, you have to determine optimal assignments of programs to memory regions. Your program is given the sizes of the memory regions available for the execution of programs, and for each program a description of how its running time depends on the amount of memory available to it. Your program has to find the execution schedule of the programs that minimizes the average turnaround time for the programs. An execution schedule is an assignment of programs to memory regions and times, such that no two programs use the same memory region at the same time, and no program is assigned to a memory region of size less than its minimum memory requirement. The turnaround time of the program is the difference between the time when the program was submitted for execution (which is time zero for all programs in this problem), and the time that the program completes execution.

Input

The input data will contain multiple test cases. Each test case begins with a line containing a pair of integers m and n. The number m specifies the number of regions into which primary memory has been partitioned ($1 \leq m \leq 10$), and n specifies the number of programs to be executed ($1 \leq n \leq 50$).

The next line contains m positive integers giving the sizes of the m memory regions. Following this are n lines, describing the time-space tradeoffs for each of the n programs. Each line starts with a positive integer k ($k \leq 10$), followed by k pairs of positive integers $s_1, t_1, s_2, t_2, \ldots, s_k, t_k$, that satisfy $s_i < s_{i+1}$ for $1 \leq i < k$. The minimum space requirement of the program is s_1, i.e. it cannot run in a partition of size less than this number. If the program runs in a memory partition of size s, where $s_i \leq s < s_{i+1}$ for some i, then its execution time will be t_i. Finally, if the programs runs in a memory partition of size s_k or more, then its execution time will be t_k.

A pair of zeroes will follow the input for the last test case.

You may assume that each program will execute in exactly the time specified for the given region size, regardless of the number of other programs in the system. No program will have a memory requirement larger than that of the largest memory region.

Output

For each test case, first display the case number (starting with 1 and increasing sequentially). Then print the minimum average turnaround time for the set of programs with two digits to the right of the decimal point. Follow this by the description of an execution schedule that achieves this average turnaround time. Display one line for each program, in the order they were given in the input, that identifies the program number, the region in which it was executed (numbered in the order given in the input), the time when the program started execution, and the time when the program completed execution. Follow the format shown in the sample output, and print a blank line after each test case.

If there are multiple program orderings or assignments to memory regions that yield the same minimum average turnaround time, give one of the schedules with the minimum average turnaround

time.

Sample Input

```
2 4
40 60
1 35 4
1 20 3
1 40 10
1 60 7
3 5
10 20 30
2 10 50 12 30
2 10 100 20 25
1 25 19
1 19 41
2 10 18 30 42
0 0
```

Sample Output

```
Case 1
Average turnaround time = 7.75
Program 1 runs in region 1 from 0 to 4
Program 2 runs in region 2 from 0 to 3
Program 3 runs in region 1 from 4 to 14
Program 4 runs in region 2 from 3 to 10

Case 2
Average turnaround time = 35.40
Program 1 runs in region 2 from 25 to 55
Program 2 runs in region 2 from 0 to 25
Program 3 runs in region 3 from 0 to 19
Program 4 runs in region 3 from 19 to 60
Program 5 runs in region 1 from 0 to 18
```