ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger

HS 2014
Exercise Sheet 6

ACM Programming Challenges Lab

Exercise 1 – *Easy exams*

Description The exam session at ETH is approaching and the dean needs your help in choosing the difficulty of the exams. The exam session consists of $1 \leq m \leq 10000$ exams, numbered from 0 to $m-1$ in the order in which they are held. In principle, each of the exams has an *expected grade* $1 \leq g_i \leq 6$, which is the average grade of the students that took the exam last year. However, the dean is thinking that some exams might be too easy or too hard. So she will give you a series of $0 \leq \ell \leq 1000$ proposed changes of the form `i g`, meaning that the i-th exam will now have an expected grade of g (instead of whatever it was before).

There are $1 \leq n \leq 100$ students at the university. Since the students are quite lazy, each one wants to register for only one exam – and this exam should be as easy as possible (if they don't register for any exams, they will be kicked out of the university, so this is not an option).

However, each student is available only available for a specified period of time, let us say from exam number a_i to exam number b_i inclusive ($0 \leq i < n$ and $a_i \leq b_i$), preferring to take off the rest of the session for more immediately appealing activities. All students have made plans already, and changing the times during which they will be available is not an option for them. Thus, each student sits the *easiest* exam that is held during the time where he/she is available. Note that an exam is easy if it has a high expected grade.

Assume that student i picks an exam with expected grade g_i. Then the *expected average grade* of the students is $(g_0 + \cdots + g_{n-1})/n$. Your task is to compute the expected average grade of the students, first for the initial assignment of expected grades, and after that for each $1 \leq i \leq \ell$ for an assignment incorporating the first i changes to difficulty.

Input The first line of the input contains the number of test cases $1 \leq t \leq 100$. The first line of each test case contains the integers n, m, ℓ, separated by spaces. This is followed by a line containing the m floating-point values $g_0, g_1, \ldots, g_{m-1}$. This is followed by n lines, each containing a pair a_i, b_i (separated by a space), representing the fact that the i-th student is available from exam a_i to exam b_i inclusive (recall that exams are numbered starting from 0). Finally, there are ℓ lines of the form `i g`, which encode the proposed changes to exam difficulty.

Output For each test case you should output $\ell + 1$ values, each on a separate line. First, output the expected average grade of the students under the original difficulties. After that, for each $1 \leq i \leq \ell$, output the expected average grade of the students after making the first i changes to the difficulties. Ouput the expected average degrees with a precision of two decimal places (without rounding).

Sample input

```
1
2 5 2
3.25 2.0 5.0 4.0 4.0
0 3
1 2
2 3.25
3 3.25
```

Sample output

```
5
3.62
3.25
```