Advanced Problem

Energy x Delay

14 March 2009

Source File	`energy.{java,c,cc}`
Input File	`energy.in`
Output File	*standard output*

Paul works for a big company called Abacus Computers and Maintenance (*ACM*). His job is to provide maintenance of computers for clients of ACM, located in different parts of the country. For this reason, Paul normally spends a good number of hours each week on flights. Obviously, Paul always takes along his laptop, so that he can work on many tasks related to his job while flying.

As batteries of laptops generally won't last long, Paul has studied alternatives to increase the duration time of the battery during flights. He found that modern processors can operate at different frequency levels, offering a compromise between performance and energy consumption. Paul's initial idea was to simply configure his laptop at the lowest frequency. However, he noted that it was not useful to simply use the lowest frequency as tasks would execute very slowly on his laptop. The would mean that he would not have sufficient time to execute all the necessary tasks, and there would be unused energy remaining in the battery at the end of the flight.

Paul noted, however, that the effect of frequency levels on performances varies from one application to another, depending on whether they are limited by memory, CPU or I/O. Additionally, as modern processors permit the frequency level to be modified by software, Paul plans to use this mechanism to increase the usage time of the battery of his laptop, while still maintaining a reasonable performance. In order to consider both energy and performance, Paul decided to use a metric already well-known, called the Energy Delay Product (*EDP*).

Paul has a list of programs that must be executed sequentially, and all the information of time and energy needed to execute each program at each frequency level, along with the information of how much energy is spent to make processor change frequency. However, to test his new idea, Paul still has a problem: like most system administrators, he doesn't like programming. He is asking for your help, for you are a great friend of his and an expert in algorithms and programming, to determine the frequency level at which each of his programs should be executed so as to minimize the total EDP.

Input

The input contains several test cases. The first line of a test case contains four integers F, P, E, and A, identifying

- the number of frequency levels supported by the processor of Paul's laptop ($1 \leq F \leq 20$),
- the number of programs to be executed sequentially ($1 \leq P \leq 5000$),
- the energy needed, in joules, to change between any two frequency levels ($1 \leq E \leq 100$), and
- the time (in milliseconds) to change between any two frequency levels ($1 \leq A \leq 100$).

respectively.

The frequency levels are identified by integers from 1 to F, and the programs are identified by integers from 1 to P.

After the first line, the following $P * F$ lines describe the programs, in P blocks of F consecutive lines. Each block of F lines corresponds to a program (the first F lines correspond to program 1, the next F lines correspond to program 2, etc.). The f^{th} line in the block for program p contains two integers

- $E_{p,f}$ the energy (in joules) execute program p at frequency level f, ($1 \leq E_{p,f} \leq 1000$), and
- $A_{p,f}$ the time (in milliseconds) to execute program p at frequency level f, $1 \leq A_{p,f} \leq 1000$.

To standardize things, assume that Paul's laptop begins each test case at frequency level 1. Between each program, a new frequency can be selected, by incurring the appropriate energy and time cost. Note that the end of input is indicated by $F = P = E = A = 0$.

Output

For each test case, your program should produce one line of output, containing the minimum EDP to sequentially execute the set of programs from 1 to P (that is, in the order they appear in the input), setting the frequency as necessary betweeen programs.

Output is emitted to standard output, with no leading or trailing spaces.

C, C++	`stdout`
C++	`cout`
Java	`System.out`

Example

Sample input and output are given in figures 1 and 2, respectively.

```
2␣3␣10␣10
50␣120
100␣90
500␣600
600␣500
400␣1000
500␣700
3␣3␣2␣5
7␣10
8␣5
15␣4
12␣4
11␣5
12␣4
7␣10
8␣5
15␣4
0␣0␣0␣0
```

Figure 1: Input

```
656100
145
```

Figure 2: Output