

PDP-27 (2015) - Camp common - 4 (permit)

Time limit: 1.0s Memory limit: 128M

PERMIT

An employee has determined which of the next T days they will drive to the office. We thus consider a function `drive` : $\{1, \dots, T\} \rightarrow \{0, 1\}$ such that `drive(i) = 1` if the employee drives to work, and `drive(i) = 0` otherwise, for $i = 1, \dots, T$. The employee can only park in the sheltered area near the office, which requires a permit. There are k different types of permits available: $(d_1, c_1), \dots, (d_k, c_k)$, where $d_1 < \dots < d_k$, $c_1 < \dots < c_k$, and permit j has a duration of d_j days and costs c_j euros, for $j = 1, \dots, k$.

We assume that the longer the duration of a permit type, the lower the cost per day (i.e., if $d_i > d_j$, then $\frac{c_i}{d_i} < \frac{c_j}{d_j}$). For example, there might be a daily permit costing 1 euro, a weekly permit costing 5 euros, a monthly permit costing 18 euros, a six-month permit costing 90 euros, and a yearly permit costing 120 euros.

Formulate and implement the most efficient algorithm possible to calculate a collection of permits with the minimum cost that covers all the days the employee drives to the office.

Input

Your program should read its input from the text file "permit.in". The first line contains T and k , the second line contains the T values of the function `drive`, and the next k lines contain the pair (duration, cost) for each permit type.

Output

The output of the program should be written to the file "permit.out". It should report the total cost of the permits purchased, on the first line.

Constraints

$$1 \leq T \leq 100.000$$

$$1 \leq K \leq 5$$

Time Limit: 1 sec.

Maximum Available Memory: 128 MB.

Example of Input - Output

STDIN (permit.in)

```
14 3
1 0 1 1 1 1 1 0 1 0 0 0 0 1
1 1
7 5
14 10
```

STDOUT (permit.out)

