

NOI '17 P5 - Vegetables

Time limit: 1.0s **Memory limit:** 256M

Description

Little N is the administrator of the vegetable warehouse and is responsible for designing the sales plan of vegetables.

In the vegetable warehouse, there are n kinds of vegetables stored in total. Little N needs to design a reasonable sales plan based on the characteristics of different vegetables and comprehensively consider various factors to obtain the most benefits.

When calculating the income from selling vegetables, for every unit of i th vegetable sold, you can get a_i income.

In particular, since the policy encourages merchants to conduct diversified sales, when selling the i th vegetable for the first time, they will also get an additional income of s_i .

At the start of the operation, the stock of vegetable i is c_i units.

However, the preservation time of vegetables is very limited, once they go bad, they cannot be sold, but the smart little N has calculated the time for each unit of vegetables to go bad: for the i th vegetable, there is a freshness value x_i , and there will be x_i units of vegetables going bad at the end of each day, until all vegetables go bad. (Note: The spoilage time of each unit of vegetables is fixed and does not change with sales)

Formally: for all positive integers d satisfying the condition $d \times x_i \leq c_i$, x_i units of vegetables will spoil at the end of d day.

In particular, if $(d - 1) \times x_i \leq c_i < d \times x_i$, then $c_i - (d - 1) \times x_i$ units of vegetables will spoil by the end of d days.

Note that when $x_i = 0$, it means that this vegetable will not go bad.

At the same time, the total amount of vegetables sold every day is also limited, and cannot exceed m units at most.

Now, Little N has k query. Each query is of the form: Given p_j , if you need to sell for p_j days, what is the maximum profit you can get?

Input Format

The first line contains three positive integers n, m, k , which respectively represent the number of types of vegetables, the upper limit of the total amount of vegetables that can be sold every day, and the number of questions raised by small N.

In the next n lines, enter four non-negative integers in each line to describe the characteristics of a vegetable, which are a_i, s_i, c_i, x_i in turn, and the meanings are as described above.

In the next k lines, enter a non-negative integer p_j in each line, the meaning is as described above.

Output Format

Output k lines, each line contains an integer, and the number in line i represents the answer to question i .

Sample Input

```
2 3 2
3 3 3 3
2 5 8 3
1
3
```

Sample Output

```
16
27
```

Explanation for Sample Output

There are two types of vegetables:

When selling the 1 vegetable, you can get 3 for each unit sold, and you can get an additional 3 when you sell this vegetable for the first time. There are 3 units of this vegetable, all spoiled by the end of the first day.

When you sell the 2 vegetable, you can get a profit of 2 for each unit sold, and when you sell this vegetable for the first time, you can get an additional profit of 5. There are 8 units of this vegetable, of which 3 units are spoiled at the end of the first day, 3 units are spoiled at the end of the second day, and 2 units are spoiled at the end of the third day.

When only selling for 1 days, 2 units of the first vegetable and 1 units of the second vegetable should be sold.

In this case: the payoff for selling the first vegetable is $2 \times 3 + 3$; the payoff for selling the second vegetable is $1 \times 2 + 5$; and the payoff in total is $(2 \times 3 + 3) + (1 \times 2 + 5) = 16$.

When only selling for 3 days, you should sell 3 units of the first vegetable on the first day, 3 units of the second vegetable on the second day (in this case choose to sell 3 units that will spoil at the end of the second day), and sell 2 units of the second vegetable on the third day.

In this case: the payoff for selling the first vegetable is $3 \times 3 + 3$; the payoff for selling the second vegetable is $(3 + 2) \times 2 + 5$; and the payoff in total is $(3 \times 3 + 3) + [(3 + 2) \times 2 + 5] = 27$.

Constraints

Test case	n	m	p_j	Property 1	Property 2
1	≤ 2	≤ 10	$\leq 10^3$	No	No
2	≤ 3	≤ 10	$\leq 10^3$	No	No
3	≤ 4	≤ 10	$\leq 10^3$	No	No
4	$\leq 10^3$	≤ 10	≤ 2	No	No
5	$\leq 10^3$	≤ 10	≤ 3	No	No
6	$\leq 10^3$	≤ 10	≤ 4	No	No
7	≤ 4	≤ 1	≤ 4	No	No
8	≤ 6	≤ 2	≤ 6	No	No
9	≤ 8	≤ 1	≤ 8	No	No
10	≤ 10	≤ 2	≤ 10	No	No
11	≤ 20	≤ 3	≤ 20	No	No
12	$\leq 10^2$	≤ 10	$\leq 10^2$	Yes	No
13	$\leq 10^2$	≤ 10	$\leq 10^2$	No	Yes
14	$\leq 10^2$	≤ 10	$\leq 10^2$	No	No
15	$\leq 10^2$	≤ 10	$\leq 10^2$	No	No
16	$\leq 10^3$	≤ 10	$\leq 10^3$	Yes	Yes
17	$\leq 10^3$	≤ 10	$\leq 10^3$	Yes	No
18	$\leq 10^3$	≤ 10	$\leq 10^3$	No	Yes
19	$\leq 10^3$	≤ 10	$\leq 10^3$	No	No
20	$\leq 10^3$	≤ 10	$\leq 10^3$	No	No
21	$\leq 10^5$	≤ 10	$\leq 10^5$	Yes	Yes
22	$\leq 10^5$	≤ 10	$\leq 10^5$	Yes	No
23	$\leq 10^5$	≤ 10	$\leq 10^5$	No	Yes
24	$\leq 10^5$	≤ 10	$\leq 10^5$	No	No
25	$\leq 10^5$	≤ 10	$\leq 10^5$	No	No

Property 1: all s_i are 0;

Property 2: All x_i are 0.

For all test data, it is guaranteed that p_j in k sets of queries are different from each other.

For all test data, it is guaranteed that $0 < a_i, c_i \leq 10^9, 0 \leq s_i, x_i \leq 10^9$.