Problem C. Common Mistake

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second

In the era of dreadroughts and battleships, people love to stack as many guns as possible onto warships to increase their fire power.

However, this is a common mistake. Adding too many guns will decrease maneuverability and mobility, and ends up decreasing the overall battle efficiency of the ships.

Given an array of N guns available, your are in charge of making the optimal selection so the battle efficiency of a ship is maximized.

You are given the following simplified model:

Each gun will have a weight w_i and power p_i.

The ship has K load ranges. While the total weight of all guns selected is in load range $[l_i, r_i]$, the battle efficiency of the ship is calculated as $m_i \sum_j p_j$ where m_i is the modifier associated with that load range, and j denotes the indices of all selected guns.

Yamato with her 9x46cm guns

Note m_i will not necessarily decrease as l_i goes up, because having more guns will also bring crew members pride and increase their morale. This in turn will bring up the battle efficiency of the whole ship.

Input

The first line of input contains one integer $N(1 \leq N \leq 200)$, the number of guns available.

In the next N lines, each line will contain two nonnegative integers, $w_i, p_i(1 \leq w_i, p_i \leq 100)$.

The next line contains one integer $K(2 \leq K \leq 200)$, the number of load ranges.

In the next K lines, each line contains two numbers l_i, m_i, the first integer is the starting weight $l_i(0 \leq l_i \leq 20000)$ of this load range. The second number is a real number, the power modifier $m_i(0 \leq m_i \leq 10)$.

It is guaranteed the load ranges will be given in the increasing order of l_i, so l_0 will always be 0. All values of l_i are distinct. It is implied that $r_i = l_{i+1} - 1$ and r_K does not matter (you can think of it as infinity).

It is also guaranteed the last load range will always have power modifier 0, which can be considered as right past the maximum load capacity of this ship.

Output

A single line with an number giving the maximum battle efficiency of the ship. The floating number is **rounded up to 3 digits after decimal points**.

Examples

stdin	stdout
3 10 30 20 40 20 40 4 0 1.0 20 2.5 50 1.8 60 0	200.000
2 100 100 200 200 2 0 2.5 10 0	0.000