4-1. Organizing the Best Squad

With the World Badminton Championships gaining popularity worldwide, Badminton World Federation is set to host the badminton All-Stars game in late September. There will be five events: men's singles, women's singles, men's doubles, women's doubles, and mixed doubles. Each participating country should form its national team for each event, and South Korea is not an exception here.

South Korean leaders decided to select the members of the team from among the players in the national badminton league. There are N players in the league, numbered 0 through $N-1$. Teams of the singles event and mixed doubles event are already formed by the results of the All-Stars selection match. However, teams for the men's doubles and the women's doubles are not chosen yet, because the two best players in the league don't necessarily play well together.

In a doubles match, one player plays the role of the attacker, and another player plays the role of defender. The Badminton Korea Association did research and quantified the attacking skill and defending skill of each player. When the attacking skill of the attacker is a, and the defending skill of the defender is d, the *skill* of the doubles team is $a+d$.

Also, the popularity of each player is an essential factor. (Note that the association is to select the teams for the all-*stars* match.) Thus, the association also quantified the popularity of each player. The *popularity* of the doubles team is the sum of popularities of each player in the team.

Some people think skills are important while others think popularity is important. Therefore, we are going to choose the importance factor of skills X and the importance factor of popularity Y, and try to form a team that maximizes the value "X $\times$ (skill of the team) $+$ Y $\times$ (popularity of the team)".

Given the information of the players and the coefficients X and Y, write a program that computes the best doubles team.

Implementation details

You should implement the following procedure and function:

```
void Init (int[] A, int[] D, int[] P)
```

- A, D, P: arrays of length N. Player i ($0 \le i \le N-1$) has attacking skill of $A[i]$, defensing skill of $D[i]$, and popularity of $P[i]$.
- This procedure is called exactly once, and before any call to `BestSquad`.

```
int64 BestSquad(int X, int Y)
```

- This function describes a request to form the best squad.
- X: the coefficient of skill
- Y: the coefficient of popularity
- This function is called Q times.
- This function should return the maximum value of "$X \times$ (skill of the team) $+ Y \times$ (popularity of the team)" among all possible double teams.

Constraints

- $2 \le N \le 300\,000$
- $1 \le Q \le 300\,000$
- $1 \le A[i], D[i], P[i] \le 1\,000\,000\,000$ (for all $0 \le i \le N-1$)
- $1 \le X, Y \le 1\,000\,000\,000$ for any call to `BestSquad`

Subtasks

1. (19 points) $Q = 1$
2. (28 points) $D[i] = 1$ for all $0 \le i \le N-1$
3. (53 points) No additional constraints.

Example

Let $N = 4$, $A = [3, 2, 2, 4]$, $D = [1, 3, 1, 4]$, $P = [4, 3, 6, 1]$, and $Q = 2$.

The grader first calls the `Init` function as below:

```
Init([3, 2, 2, 4], [1, 3, 1, 4], [4, 3, 6, 1])
```

Then, the grader calls `BestSquad` $Q = 2$ times. Consider the following call:

```
BestSquad(1, 3)
```

The answer is 34.

Consider the following call:

```
BestSquad(3, 1)
```

The answer is 26.

Sample grader

You can download the sample grader package on the same page you downloaded the problem statement. (scroll down if you don't see the attachment)

If you use IDEs like Visual Studio, Eclipse or Code::Blocks, then import `squad.cpp`, `squad.h` and `grader.cpp` into one project and you will be able to compile all these files at once.

If you want to compile by yourself, refer to the compilation commands in the statement page.

You should submit only `squad.cpp`.

Input format

- line 1: N
- line $2 + i$ ($0 \leq i \leq N - 1$): $A[i]$ $D[i]$ $P[i]$
- line $N + 2$: Q
- line $N + 3 + j$ ($0 \leq j \leq Q - 1$): $X[j]$ $Y[j]$

Here, $X[j]$ and $Y[j]$ are parameters for the call of `BestSquad` for the request j.

Output format

- line $1 + j$ ($0 \leq j \leq Q - 1$): the return value of `BestSquad` for the request j.