C. Save the Nature

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are an environmental activist at heart but the reality is harsh and you are just a cashier in a cinema. But you can still do something!

You have n tickets to sell. The price of the i-th ticket is p_i. As a teller, you have a possibility to select the order in which the tickets will be sold (i.e. a permutation of the tickets). You know that the cinema participates in two ecological restoration programs applying them **to the order you chose**:

- The $x\%$ of the price of each the a-th sold ticket (a-th, $2a$-th, $3a$-th and so on) *in the order you chose* is aimed for research and spreading of renewable energy sources.
- The $y\%$ of the price of each the b-th sold ticket (b-th, $2b$-th, $3b$-th and so on) *in the order you chose* is aimed for pollution abatement.

If the ticket is in both programs then the $(x + y)\%$ are used for environmental activities. Also, it's known that all prices are multiples of 100, so there is no need in any rounding.

For example, if you'd like to sell tickets with prices $[400, 100, 300, 200]$ and the cinema pays 10% of each 2-nd sold ticket and 20% of each 3-rd sold ticket, then arranging them in order $[100, 200, 300, 400]$ will lead to contribution equal to $100 \cdot 0 + 200 \cdot 0.1 + 300 \cdot 0.2 + 400 \cdot 0.1 = 120$. But arranging them in order $[100, 300, 400, 200]$ will lead to $100 \cdot 0 + 300 \cdot 0.1 + 400 \cdot 0.2 + 200 \cdot 0.1 = 130$.

Nature can't wait, so you decided to change the order of tickets in such a way, so that the **total** contribution to programs will reach at least k in **minimum** number of sold tickets. Or say that it's impossible to do so. In other words, find the minimum number of tickets which are needed to be sold in order to earn at least k.

Input

The first line contains a single integer q ($1 \le q \le 100$) — the number of independent queries. Each query consists of 5 lines.

The first line of each query contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of tickets.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($100 \le p_i \le 10^9$, $p_i \bmod 100 = 0$) — the corresponding prices of tickets.

The third line contains two integers x and a ($1 \le x \le 100$, $x + y \le 100$, $1 \le a \le n$) — the parameters of the first program.

The fourth line contains two integers y and b ($1 \le y \le 100$, $x + y \le 100$, $1 \le b \le n$) — the parameters of the second program.

The fifth line contains single integer k ($1 \le k \le 10^{14}$) — the required total contribution.

It's guaranteed that the total number of tickets per test doesn't exceed $2 \cdot 10^5$.

Output

Print q integers — one per query.

For each query, print the minimum number of tickets you need to sell to make the total ecological contribution of at least k if you can sell tickets in any order.

If the total contribution can not be achieved selling all the tickets, print -1.

Example

input Copy

```
4
1
100
50 1
49 1
100
8
100 200 100 200 100 200 100 100
10 2
15 3
107
3
1000000000 1000000000 1000000000
50 1
50 1
3000000000
5
200 100 100 100 100
69 5
31 2
90
```

output Copy

```
-1
6
3
4
```

- Announcement #2 (ru)
- Tutorial (en)

Note

In the first query the total contribution is equal to $50 + 49 = 99 < 100$, so it's impossible to gather enough money.

In the second query you can rearrange tickets in a following way: $[100, 100, 200, 200, 100, 200, 100, 100]$ and the total contribution from the first 6 tickets is equal to $100 \cdot 0 + 100 \cdot 0.1 + 200 \cdot 0.15 + 200 \cdot 0.1 + 100 \cdot 0 + 200 \cdot 0.25 = 10 + 30 + 20 + 50 = 110$.

In the third query the full price of each ticket goes to the environmental activities.

In the fourth query you can rearrange tickets as $[100, 200, 100, 100, 100]$ and the total contribution from the first 4 tickets is $100 \cdot 0 + 200 \cdot 0.31 + 100 \cdot 0 + 100 \cdot 0.31 = 62 + 31 = 93$.