B. Fancy Coins

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Monocarp is going to make a purchase with cost of exactly m burles.

He has two types of coins, in the following quantities:

- coins worth 1 burle: a_1 regular coins and infinitely many fancy coins;
- coins worth k burles: a_k regular coins and infinitely many fancy coins.

Monocarp wants to make his purchase in such a way that there's **no change** — the total worth of provided coins is **exactly** m. He can use both regular and fancy coins. However, he wants to spend as little fancy coins as possible.

What's the smallest total number of fancy coins he can use to make a purchase?

Input

The first line contains a single integer t ($1 \le t \le 3 \cdot 10^4$) — the number of testcases.

The only line of each testcase contains four integers m, k, a_1 and a_k ($1 \le m \le 10^8$; $2 \le k \le 10^8$; $0 \le a_1, a_k \le 10^8$) — the cost of the purchase, the worth of the second type of coin and the amounts of regular coins of both types, respectively.

Output

For each testcase, print a single integer — the smallest total number of fancy coins Monocarp can use to make a purchase.

Example

input Copy

```
4
11 3 0 0
11 3 20 20
11 3 6 1
100000000 2 0 0
```

output Copy

```
5
0
1
50000000
```

Note

In the first testcase, there are no regular coins of either type. Monocarp can use 2 fancy coins worth 1 burle and 3 fancy coins worth 3 (since $k = 3$) burles to get 11 total burles with 5 total fancy coins.

In the second testcase, Monocarp has a lot of regular coins of both types. He can use 11 regular coins worth 1 burle, for example. Notice that Monocarp doesn't have to minimize the total number of used coins. That way he uses 0 fancy coins.

In the third testcase, Monocarp can use 5 regular coins worth 1 burle and 1 regular coin worth 3 burles. That will get him to 8 total burles when he needs 11. So, 1 fancy coin worth 3 burles is enough.