A. Three Piles of Candies

Alice and Bob have received three big piles of candies as a gift. Now they want to divide these candies as fair as possible. To do this, Alice takes one pile of candies, then Bob takes one of the other two piles. The last pile is split between Alice and Bob as they want: for example, it is possible that Alice takes the whole pile, and Bob gets nothing from it.

After taking the candies from the piles, if Alice has more candies than Bob, she discards some candies so that the number of candies she has is equal to the number of candies Bob has. Of course, Bob does the same if he has more candies.

Alice and Bob want to have as many candies as possible, and they plan the process of dividing candies accordingly. Please calculate the maximum number of candies Alice can have after this division process (of course, Bob will have the same number of candies).

You have to answer q independent queries.

Let's see the following example: $[1, 3, 4]$. Then Alice can choose the third pile, Bob can take the second pile, and then the only candy from the first pile goes to Bob — then Alice has 4 candies, and Bob has 4 candies.

Another example is $[1, 10, 100]$. Then Alice can choose the second pile, Bob can choose the first pile, and candies from the third pile can be divided in such a way that Bob takes 54 candies, and Alice takes 46 candies. Now Bob has 55 candies, and Alice has 56 candies, so she has to discard one candy — and after that, she has 55 candies too.

Input

The first line of the input contains one integer q ($1 \le q \le 1000$) — the number of queries. Then q queries follow.

The only line of the query contains three integers a, b and c ($1 \le a, b, c \le 10^{16}$) — the number of candies in the first, second and third piles correspondingly.

Output

Print q lines. The i-th line should contain the answer for the i-th query — the maximum number of candies Alice can have after the division, if both Alice and Bob act optimally (of course, Bob will have the same number of candies).

Example

input

```
4
1 3 4
1 10 100
10000000000000000 10000000000000000 10000000000000000
23 34 45
```

output

```
4
55
15000000000000000
51
```