

A. Frog Jumping

time limit per test

1 second

memory limit per test

256 megabytes

input

standard input

output

standard output

A frog is currently at the point 0 on a coordinate axis Ox . It jumps by the following algorithm: the first jump is a units to the right, the second jump is b units to the left, the third jump is a units to the right, the fourth jump is b units to the left, and so on.

Formally:

- if the frog has jumped an even number of times (before the current jump), it jumps from its current position x to position $x+a$;
- otherwise it jumps from its current position x to position $x-b$.

Your task is to calculate the position of the frog after k jumps.

But... One more thing. You are watching t different frogs so you have to answer t independent queries.

Input

The first line of the input contains one integer t ($1 \leq t \leq 1000$) — the number of queries.

Each of the next t lines contain queries (one query per line).

The query is described as three space-separated integers a, b, k ($1 \leq a, b, k \leq 10^9$) — the lengths of two types of jumps and the number of jumps, respectively.

Output

Print t integers. The i -th integer should be the answer for the i -th query.

Example

input

Copy

6

5 2 3

100 1 4

1 10 5

1000000000 1 6

1 1 1000000000

1 1 999999999

output

Copy

8

198

-17

2999999997

0

1

Note

In the first query frog jumps 5 to the right, 2 to the left and 5 to the right so the answer is $5-2+5=8$.

In the second query frog jumps 100 to the right, 1 to the left, 100 to the right and 1 to the left so the answer is $100-1+100-1=198$.

In the third query the answer is $1-10+1-10+1=-17$.

In the fourth query the answer is $10^9-1+10^9-1+10^9-1=2999999997$.

In the fifth query all frog's jumps are neutralized by each other so the answer is 0.

The sixth query is the same as the fifth but without the last jump so the answer is 1.