

Task 1:

Little Bobby loves chocolate. He frequently goes to his favorite store, Penny Shop, to buy them. They are having a promotion at Penny Shop. If Bobby saves enough wrappers, he can turn them in for a free chocolate.

Example:

$$n = 15$$

$$c = 3$$

$$m = 2$$

He has 15 dollar to spend, each bar costs 3 dollar, and he can turn in 2 wrappers to receive another bar.

Now, Initially, he buys 5 bars and has 5 wrappers after eating them. He turns in 4 wrappers, leaving him with 1, for 2 more bars. After eating those two, he has 3 wrappers, turns in 2, leaving him with 1 wrapper and his new bar. Once he eats that one bar, he has 2 wrappers and turns them in for another bar. After eating that one, he only has 1 wrapper, and his feast ends. Overall, he has eaten $5 + 2 + 1 + 1 = 9$ bars.

Now, create a function to calculate how many bars can Bobby consume.

Function Description:

Complete the *chocolateFeast* function as describe below:

chocolateFeast has the following parameter(s):

- int *n*: Bobby's initial amount of money
- int *c*: the cost of a chocolate bar
- int *m*: the number of wrappers he can turn in for a free bar

Returns:

- int *total_bars*: the number of chocolates Bobby can eat after taking full advantage of the promotion

Note:

Little Bobby will always turn in his wrappers if he has enough to

get a free chocolate.

Input Format:

The first line contains an integer, t , the number of test cases to analyze.

Each of the next t lines contains three space-separated integers: n, c, m . They represent money to spend, cost of a chocolate, and the number of wrappers he can turn in for a free chocolate.

Constraints:

- $1 \leq t \leq 1000$
- $2 \leq n \leq 10^5$
- $1 \leq c \leq n$
- $2 \leq m \leq n$

Output Format:

Prints the *total_bars* for each test cases.

Sample Input	Sample Output
2 10 2 5 12 4 4	6 3