

A. Optimal Currency Exchange

time limit per test: 1.5 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Andrew was very excited to participate in Olympiad of Metropolises. Days flew by quickly, and Andrew is already at the airport, ready to go home. He has n rubles left, and would like to exchange them to euro and dollar bills. Andrew can mix dollar bills and euro bills in whatever way he wants. The price of one dollar is d rubles, and one euro costs e rubles.

Recall that there exist the following dollar bills: 1, 2, 5, 10, 20, 50, 100, and the following euro bills — 5, 10, 20, 50, 100, 200 (note that, in this problem we do **not** consider the 500 euro bill, it is hard to find such bills in the currency exchange points). Andrew can buy any combination of bills, and his goal is to minimize the total number of rubles he will have after the exchange.

Help him — write a program that given integers n , e and d , finds the minimum number of rubles Andrew can get after buying dollar and euro bills.

Input

The first line of the input contains one integer n ($1 \leq n \leq 10^8$) — the initial sum in rubles Andrew has.

The second line of the input contains one integer d ($30 \leq d \leq 100$) — the price of one dollar in rubles.

The third line of the input contains integer e ($30 \leq e \leq 100$) — the price of one euro in rubles.

Output

Output one integer — the minimum number of rubles Andrew can have after buying dollar and euro bills optimally.

Examples

input	Copy
100 60 70	
output	Copy
40	

input	Copy
410 55 70	
output	Copy
5	

input	Copy
600 60 70	
output	Copy
0	

Note


In the first example, we can buy just 1 dollar because there is no 1 euro bill.

In the second example, optimal exchange is to buy 5 euro and 1 dollar.

Codeforces Round #583 (Div. 1 + Div. 2, based on Olympiad of Metropolises)

Finished

Practice



[→ Virtual participation](#)

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

[→ Practice](#)

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

[→ Clone Contest to Mashup](#)

You can clone this contest to a mashup.

Clone Contest

[→ Submit?](#)

Language: GNU G++11 5.1.0

Choose file:

选择文件

 未选择任何文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

[→ Problem tags](#)

brute force

math

*1300

No tag edit access

[→ Contest materials](#)

Announcement #1 (en)

Announcement #2 (ru)

Tutorial (en)

In the third example, optimal exchange is to buy 10 dollars in one bill.

[Codeforces](#) (c) Copyright 2010-2019 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Sep/10/2019 15:04:24^{UTC+8} (e1).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



ITMO UNIVERSITY