

A. Polycarp and Coins

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Polycarp must pay **exactly** n burles at the checkout. He has coins of two nominal values: 1 burle and 2 burles. Polycarp likes both kinds of coins equally. So he doesn't want to pay with more coins of one type than with the other.

Thus, Polycarp wants to minimize the difference between the count of coins of 1 burle and 2 burles being used. Help him by determining two non-negative integer values c_1 and c_2 which are the number of coins of 1 burle and 2 burles, respectively, so that the total value of that number of coins is **exactly** n (i. e. $c_1 + 2 \cdot c_2 = n$), and the absolute value of the difference between c_1 and c_2 is as little as possible (i. e. you must minimize $|c_1 - c_2|$).

Input

The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases. Then t test cases follow.

Each test case consists of one line. This line contains one integer n ($1 \leq n \leq 10^9$) — the number of burles to be paid by Polycarp.

Output

For each test case, output a separate line containing two integers c_1 and c_2 ($c_1, c_2 \geq 0$) separated by a space where c_1 is the number of coins of 1 burle and c_2 is the number of coins of 2 burles. If there are multiple optimal solutions, print any one.

Example

input	Copy
6 1000 30 1 32 1000000000 5	
output	Copy
334 333 10 10 1 0 10 11 333333334 333333333 1 2	

Note

The answer for the first test case is "334 333". The sum of the nominal values of all coins is $334 \cdot 1 + 333 \cdot 2 = 1000$, whereas $|334 - 333| = 1$. One can't get the better value because if $|c_1 - c_2| = 0$, then $c_1 = c_2$ and $c_1 \cdot 1 + c_1 \cdot 2 = 1000$, but then the value of c_1 isn't an integer.

The answer for the second test case is "10 10". The sum of the nominal values is $10 \cdot 1 + 10 \cdot 2 = 30$ and $|10 - 10| = 0$, whereas there's no number having an absolute value less than 0.

Codeforces Round #734 (Div. 3)

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 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](#)

→ Clone Contest to Mashup

You can clone this contest to a mashup.

[Clone Contest](#)

→ Submit?

Language: GNU G++17 7.3.0

 Choose file: [Choose File](#) No file chosen



[Submit](#)

→ Problem tags

greedy math *800

No tag edit access

→ Contest materials

- Announcement 
- Tutorial 



[Codeforces](#) (c) Copyright 2010-2021 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Sep/30/2021 12:53:04^{UTC-7} (j3).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



ITMO UNIVERSITY