A. Mystic Waves

In the land of Elaria, a young mage named Nia experiments with a magical energy value **x**.

She casts a spell that produces a sequence of **n** waves of energy — alternating between **x** and **$-x$**, starting with **x**.

For example, if **$n = 5$**, the sequence of energies will be: **$x, -x, x, -x, x$**

Nia wants to know the total magical energy after all **n** waves.

Input

The first line contains an integer **t** ($1 \le t \le 100$) — the number of test cases.

Each of the following **t** lines contains two integers **x** and **n** ($1 \le x, n \le 10$).

Output

For each test case, output a single integer — the total energy after all **n** waves.

Example

Input	**Output**
4	
1 4	0
2 5	2
3 6	0
4 7	4

B. CargoCraft Fleet

In the futuristic world of **Aerion**, the **CargoCraft** company operates a fleet of transport vehicles designed to carry supplies between colonies.

There are **two types of crafts** in the fleet:

- **Type A** crafts have **4 propulsion units**,
- **Type B** crafts have **6 propulsion units**.

You receive a report that the fleet's crafts together have **n propulsion units** in total. However, the report doesn't specify how many crafts of each type there are.

Your task is to determine **how many crafts** the fleet could possibly have — that is, the **minimum** and **maximum** number of crafts that can make up a total of **n propulsion units**.

If the total **n** cannot be made using only crafts of Type A and Type B, report that it's **impossible**.

Input

The first line contains an integer **t** (1 ≤ **t** ≤ 1000) — the number of test cases.

Each of the next **t** lines contains one integer **n** ($1 \le n \le 10^{18}$) — the total number of propulsion units.

Output

For each test case, print one line containing two integers **x** and **y** — the **minimum** and **maximum** possible number of crafts in the fleet.

If no valid combination exists, print **-1**.

Limitation

Time limit per test: 1 second memory

Limit per test: 256 megabytes

Example

Input	**Output**
4	
4	1 1
7	-1
24	4 6
998244353998244352	166374058999707392 249561088499561088

Explanation

- In the first test case, **n = 4**, meaning there is exactly one Type A craft.
- In the second, **n = 7** can't be made from 4s and 6s — hence impossible.
- In the third, **n = 24**, the fleet could be:
    - 4 Type B crafts (6×4=24),
    - 3 Type A + 2 Type B (3×4 + 2×6 = 24),
    - or 6 Type A crafts (6×4=24).

      Thus, the **minimum = 4**, **maximum = 6**.

C. Frontend Test — Login Page

Goal:
Build a simple **login page** using **HTML, CSS, and JavaScript** (no backend). You can use any frontend framework if you like (e.g. React, Vue, or plain JS).

Requirements

1. Page Layout

- Centered login form on a plain background.
- The form includes:
  - **Email** input
  - **Password** input
  - **Login** button
  - Optional "Forgot password?" link
- Responsive design (should look good on desktop & mobile).

Validation Rules

Email field:

- Must not be empty.
- Show an error if the email doesn't exist (you can simulate a small list of valid emails in the code, e.g. ["test@example.com"]).

Password field:

- Must be between **8–16 characters**.
- Must contain at least:
  - one **uppercase letter**
  - one **lowercase letter**
  - one **number**
  - one **symbol**
- Show an error if it doesn't meet the criteria.

- Show an error if the email exists, but the password is incorrect.

Successful Login

When both fields are valid and credentials match:

- Hide the login form.
- Display a **welcome message** like:
  Welcome, [user email]!
- Include a **Logout** button that returns to the login page.

Technical Notes

- All functionality must happen on the frontend (no backend calls).
- You can hardcode sample valid credentials in your code.
- Use clean, readable code with comments.
- Styling should be neat and consistent (bonus for modern design).

Submission

- Upload your code to **GitHub**.
- Share your **GitHub repository link**.
- This test contains **three tasks**, organized into folders **A**, **B**, and **C**. Each folder includes one question — please complete **all three**.
- **Language Requirements:**
    - **Question A:** Accept **Python, Java, C++, C**
    - **Question B:** Accept **Python, Java, C++, C**
    - **Question C:** Accept **Vue**, **React**, **Next.js, Go**