

Bài 1: Kiểm tra hai chuỗi có là Đảo ngữ (Anagram)

Mô tả: Nhập vào hai chuỗi ký tự. Kiểm tra xem chuỗi này có phải là bản sắp xếp lại các ký tự của chuỗi kia hay không (ví dụ: "listen" và "silent").

- **Input:** Hai chuỗi ký tự riêng biệt.
 - **Output:** "Là đảo ngữ" hoặc "Không phải đảo ngữ".
 - **Ví dụ:**
 - **Input 1:** heart, **Input 2:** earth
 - **Output:** Là đảo ngữ
 - **Giải thích:** Cả hai từ đều chứa các chữ cái giống hệt nhau (a, e, h, r, t) với cùng số lượng, chỉ khác thứ tự.
-

Bài 2: Nén chuỗi cơ bản (Run-Length Encoding)

Mô tả: Nhập vào một chuỗi có các ký tự lặp lại liên tiếp. Hãy nén chuỗi đó theo quy tắc: [ký tự][số lần lặp].

- **Input:** Một chuỗi (ví dụ: aaabbccccc).
 - **Output:** a3b2c4
 - **Ví dụ:**
 - **Input:** aabbba
 - **Output:** a2b4a1
 - **Giải thích:** Chữ 'a' xuất hiện 2 lần liên tiếp, sau đó là 4 chữ 'b', cuối cùng là 1 chữ 'a'.
-

Bài 3: Kiểm tra số Hoàn hảo (Perfect Number)

Mô tả: Nhập vào một số nguyên dương \$n\$. Kiểm tra xem đó có phải là số hoàn hảo hay không. Số hoàn hảo là số có tổng các ước thực sự của nó (không kể chính nó) bằng chính nó.

- **Input:** Một số nguyên \$n\$.
 - **Output:** "Là số hoàn hảo" hoặc "Không phải số hoàn hảo".
 - **Ví dụ:**
 - **Input:** 28
 - **Output:** Là số hoàn hảo
 - **Giải thích:** Các ước của 28 (không kể 28) là 1, 2, 4, 7, 14. Tổng: \$1+2+4+7+14 = 28\$.
-

Bài 4: Loại bỏ phần tử trùng lặp nhưng giữ nguyên thứ tự

Mô tả: Nhập vào một dãy số hoặc một chuỗi. Hãy loại bỏ các phần tử đã xuất hiện trước đó nhưng phải giữ nguyên thứ tự xuất hiện ban đầu của các phần tử còn lại.

- **Input:** Một chuỗi (ví dụ: `google`).
 - **Output:** `gole`
 - **Ví dụ:**
 - **Input:** `1, 2, 2, 3, 1, 4`
 - **Output:** `1, 2, 3, 4`
 - **Giải thích:** Chữ 'o' thứ hai và các số 2, 1 xuất hiện sau đã bị loại bỏ, những số đầu tiên được giữ lại đúng vị trí.
-

Bài 5: Bài toán Tìm cặp số có tổng bằng K (Two Sum)

Mô tả: Nhập vào một dãy số nguyên (cách nhau bởi khoảng trắng) và một số mục tiêu \$K\$. Tìm cặp số đầu tiên trong dãy có tổng bằng \$K\$.

- **Input:** Dãy số: `2 7 11 15`, \$K = 9\$.
 - **Output:** `(2, 7)`
 - **Ví dụ:**
 - **Input:** Dãy: `14 3 2`, K: `5`
 - **Output:** `(1, 4)`
 - **Giải thích:** Trong dãy số, $1 + 4 = 5$, đây là cặp số đầu tiên thỏa mãn điều kiện.
-

Bài 6: Chuyển đổi số La Mã sang số nguyên

Mô tả: Nhập vào một chuỗi ký tự đại diện cho số La Mã (I, V, X, L, C, D, M) và chuyển nó thành số nguyên tương ứng.

- **Quy tắc:** I=1, V=5, X=10, L=50, C=100, D=500, M=1000.
 - **Input:** `XIV`
 - **Output:** `14`
 - **Ví dụ:**
 - **Input:** `MIX`
 - **Output:** `1009`
 - **Giải thích:** $M(1000) + IX(9) = 1009$.
-

Bài 7: Tìm hàng có tổng lớn nhất trong Ma trận

Mô tả: Nhập vào số hàng, số cột và các giá trị của một ma trận. Hãy cho biết hàng nào có tổng các phần tử lớn nhất.

- **Input:** Ma trận 2x2: `[[1, 2], [3, 4]]`.
 - **Output:** Hàng 2 (Tổng: 7)
 - **Ví dụ:**
 - **Input:** Hàng 1: 5 5, Hàng 2: 12
 - **Output:** Hàng 1(Tổng: 10)
 - **Giải thích:** Hàng 1 có $5+5=10\$$, hàng 2 có $1+2=3\$$. Vậy hàng 1 lớn nhất.
-

Bài 8: Kiểm tra tính hợp lệ của dấu ngoặc

Mô tả: Nhập vào một chuỗi chỉ gồm các ký tự ngoặc `()`, `[]`, `{}`. Kiểm tra xem các dấu ngoặc có được đóng mở đúng quy tắc và thứ tự hay không.

- **Input:** `{[()]}{}`
 - **Output:** Hợp lệ
 - **Ví dụ:**
 - **Input:** `{[()]}{}`
 - **Output:** Không hợp lệ
 - **Giải thích:** Dấu `[` phải được đóng bởi `]` trước khi đóng `}`, ở ví dụ sai, thứ tự đóng mở bị đan xen sai quy cách.
-

Bài 9: Tìm tiền tố chung dài nhất (Longest Common Prefix)

Mô tả: Nhập vào một danh sách các từ. Tìm chuỗi ký tự là tiền tố (phần đầu) chung dài nhất mà tất cả các từ đều có.

- **Input:** `flower, flow, flight`
 - **Output:** `f`
 - **Ví dụ:**
 - **Input:** `dog, racecar, car`
 - **Output:** (Chuỗi rỗng)
 - **Giải thích:** Trong từ "flower", "flow", "flight", cả ba đều bắt đầu bằng "fl". Ở ví dụ sau, không có chữ cái đầu nào chung cho cả 3 từ.
-

Bài 10: Sắp xếp danh sách theo tần suất xuất hiện

Mô tả: Nhập vào một dãy số. Hãy sắp xếp lại dãy số sao cho số nào xuất hiện nhiều lần hơn thì đứng trước. Nếu số lần xuất hiện bằng nhau, số nhỏ hơn đứng trước.

- **Input:** 1, 1, 2, 2, 2, 3

- **Output:** 2, 2, 2, 1, 1, 3

- **Ví dụ:**

- **Input:** 4, 4, 5, 5, 6

- **Output:** 4, 4, 5, 5, 6 (Vì 4 và 5 cùng xuất hiện 2 lần, 4 nhỏ hơn đứng trước)

- **Giải thích:** Số 2 xuất hiện 3 lần (nhiều nhất) nên lên đầu, số 1 xuất hiện 2 lần, số 3 xuất hiện 1 lần.