

ĐỀ THI KIỂM TRA GIỮA KỲ
MÔN: CƠ SỞ LẬP TRÌNH
THỜI GIAN: 60 phút

Bài 1: Tính toán biểu thức (2 điểm)

Viết một chương trình Python để:

1. Cho người dùng nhập vào một số nguyên a và một số nguyên b.
2. Kiểm tra xem cả hai số a và b đều có phải là số dương hay không.
3. Nếu cả hai số đều dương, tính và in ra giá trị của biểu thức:

$$\sqrt{a^2 + b^2}$$

4. Ngược lại (nếu một trong hai hoặc cả hai số không dương), in ra một thông báo lỗi phù hợp.

Bài 2: Xử lý chuỗi và từ điển (4 điểm)

Yêu cầu:

1. Nhập vào một chuỗi gồm nhiều câu (có thể chứa dấu chấm, dấu phẩy, ký tự đặc biệt).
2. Chuẩn hóa chuỗi:
 - ✓ Chuyển tất cả thành chữ thường.
 - ✓ Loại bỏ các ký tự không phải chữ cái hoặc khoảng trắng.
3. Tách chuỗi thành danh sách các từ.
4. Đếm tần suất xuất hiện của từng từ và lưu kết quả vào một dict.
5. In ra top 5 từ xuất hiện nhiều nhất, theo định dạng: Từ: | Số lần: <số lần>

Ví dụ: Python is great! Python is easy, and Python is powerful.

Output Từ: python | Số lần: 3
 Từ: is | Số lần: 3
 Từ: great | Số lần: 1

Bài 3: Thuật toán & xử lý dữ liệu thời gian (4 điểm)

Giả sử bạn đang xây dựng hệ thống phân tích dữ liệu giao dịch.

Cho một danh sách các giao dịch theo dạng sau:

```
transactions = [ {'id': 1, 'user': 'Alice', 'amount': 200, 'time': '2025-10-28 09:10'},  
                 {'id': 2, 'user': 'Bob', 'amount': 500, 'time': '2025-10-28 09:30'},  
                 {'id': 3, 'user': 'Alice', 'amount': 700, 'time': '2025-10-28 10:00'},  
                 {'id': 4, 'user': 'Bob', 'amount': 300, 'time': '2025-10-28 10:15'},  
                 {'id': 5, 'user': 'Alice', 'amount': 400, 'time': '2025-10-28 11:00'} ]
```

Hãy viết chương trình thực hiện các chức năng sau:

1. Tính tổng số tiền giao dịch của từng người dùng.
→ Output: {'Alice': 1300, 'Bob': 800}
2. Tìm người có tổng giao dịch cao nhất.
→ Output: Alice
3. Phân tích theo khung thời gian: Nhóm các giao dịch theo giờ (09h, 10h, 11h,...), tính tổng tiền trong mỗi khung giờ.
→ Output: 09h: 700 10h: 1000 11h: 400
4. Viết hàm `detect_anomaly(transactions, threshold)`: Trả về danh sách các giao dịch có `amount > threshold` hoặc gấp 3 lần trung bình của người đó.
Ví dụ: `detect_anomaly(transactions, 600)`
→ [{'id': 3, 'user': 'Alice', 'amount': 700, 'time': '2025-10-28 10:00'}]