

Bài 1:

Cho bài toán như sau: Giả sử bạn đang phát triển một hệ thống quản lý đơn hàng cho một cửa hàng trực tuyến. Hệ thống cần tính toán tổng chi phí cho một đơn hàng dựa trên các yếu tố sau:

1. Thông tin đơn hàng:

- **Danh sách sản phẩm:** Mỗi sản phẩm có giá, số lượng, và có thể có giảm giá.
- **Phí vận chuyển:** Tính thêm phí nếu tổng giá trị đơn hàng nhỏ hơn 1 triệu.
- **Loại khách hàng:** Khách hàng thường xuyên có chiết khấu 10% trên tổng giá trị đơn hàng.

2. Yêu cầu bài toán:

- Viết hàm `TinhChiPhi(Order order)` để tính tổng chi phí của đơn hàng sau khi đã áp dụng các quy tắc giảm giá, chiết khấu và vận chuyển.
- **Order** gồm:
 - **Danh sách các sản phẩm:** Mỗi sản phẩm gồm giá gốc, số lượng và phần trăm giảm giá sản phẩm đó.
 - **IsRegularCustomer:** Khách hàng thường xuyên hay không.
 - **ShippingFee:** Phí vận chuyển.

3. Quy tắc:

- Có 1 function tính tổng giá thành khi áp dụng giảm giá và 1 function không áp dụng giảm giá.
- Nếu giá trị đơn hàng trước khi áp dụng giảm giá các sản phẩm ≥ 1 triệu, miễn phí vận chuyển.
- Nếu khách hàng là khách hàng thường xuyên, chiết khấu 10% trên tổng giá trị đơn hàng sau khi áp dụng các giảm giá của sản phẩm.

Yêu cầu:

1. Viết mã giả cho bài toán trên.
2. Dựa vào mã giả của bạn, ta sẽ áp dụng Unit test, White box test, Black box test cho các phần nào của bài toán. Nếu áp dụng White box test/ Black box test thì ta cần sinh các test có đặc điểm như thế nào?

Bài 2: Dãy con có tổng lớn nhất:

Cho dãy số nguyên a_1, a_2, \dots, a_n , một dãy con liên tiếp của dãy a là $a_l + a_{l+1} + \dots + a_r$. Hãy tìm dãy con liên tiếp có tổng lớn nhất, hay tìm một cặp (l, r) mà $1 \leq l \leq r \leq n$ có $(a_l + a_{l+1} + \dots + a_r)$ là lớn nhất.

Input:

- Dòng đầu chứa số nguyên dương $n (n \leq 10^5)$.
- Dòng thứ hai chứa n số $a_1, a_2, \dots, a_n (|a_i| \leq 10^4)$.

Output:

- Ghi ra tổng lớn nhất tìm được.

Dưới đây là 1 test ví dụ:

Example

Test 1

Input

10
3 -1 -3 5 2 5 0 -1 5 4

Copy

Output

20

Copy

Yêu cầu:

1. Viết code trâu có độ phức tạp $O(n^2)$ hoặc $O(n^3)$.
2. Viết code full có độ phức tạp $O(n)$ hoặc $O(n * \log(n))$.
3. Viết trình sinh và so test giữa “code trâu” và “code full”.
4. Trình bày các trường hợp đặc biệt về testcase của bài toán vào báo cáo.