

THỜI GIAN: 150 PHÚT
CHÚ Ý: SINH VIÊN ĐỌC KĨ ĐỂ NHÀM TÌM RA
CẤU TRÚC DỮ LIỆU PHÙ HỢP NHẤT CHO CÁC NÚT CỦA CÂY
TÌM KIẾM NHỊ PHÂN

Bài 1 (4 điểm): Viết chương trình C nhận đầu vào là các dòng liệt kê các giao dịch giữa hai con người bất kỳ với nhau trên một sàn thương mại điện tử. Mỗi dòng có nội dung như sau:
DD-MM-YYYY seller buyer productID number cost

Chẳng hạn:

24-02-2023 john peter 45E1 12 360

Dòng trên có nghĩa là vào ngày 24 tháng 02 năm 2023, john đã bán cho peter sản phẩm có mã 45E1 với số lượng 12 cái và tổng tiền là 360 Yen.

Khi người dùng nhấn dấu \$ thì chương trình sẽ ngừng việc nhập của người dùng lại.

a) Hãy kiểm tra xem trong số các dòng ở trên, liệu có bao nhiêu dòng thoả mãn yêu cầu về số tiền hợp lệ. Để đơn giản, ta giả sử không có ai thanh toán số tiền dưới 1 Yên và giao dịch trên sàn thương mại này không được vượt quá 200 Man (**2 điểm**). Nếu có 7 dòng hợp lệ thì in ra số "7"

b) Hãy kiểm tra xem trong số các dòng ở trên, liệu có bao nhiêu dòng thoả mãn yêu cầu về số sản phẩm hợp lệ: phải là số nguyên dương, và bé hơn 100 (**1 điểm**). Nếu có 7 dòng hợp lệ thì in ra số "7"

c) Hãy kiểm tra xem trong các dòng đó, dòng nào có số tiền trên một sản phẩm cao nhất. (**1 điểm**). Dòng đầu tiên được coi là dòng số 0, nếu dòng số 7 hợp lệ thì in ra số "7"

Ở bài này, trong hàm main, SV viết vòng lặp while để người dùng nhập vào từng dòng mong muốn, với mỗi dòng nhập xong thì sẽ có ba hàm (để giải quyết ba nhiệm vụ của các câu (a), (b), (c) ở trên) được gọi.

Bài 2 (4 điểm):

Với N dòng dữ liệu hợp lệ đầu tiên do người dùng nhập vào (tức hàm ở câu (a), (b) đều trả về số lớn hơn hoặc bằng N). Hãy xây dựng một cấu trúc cây tìm kiếm nhị phân với tiêu chí đánh giá hai nút (mỗi nút trong cây ứng với một dòng) xem nút nào lớn hơn:

nếu nút A có tổng số tiền lớn hơn nút B thì ta nói $A > B$

khi mà số tiền bằng nhau thì xét xem liệu số sản phẩm của A có ít hơn số sản phẩm của B không, nếu có, ta nói $A > B$

khi mà cả số sản phẩm cũng bằng nhau thì xét xem liệu tên sản phẩm A có đứng sau B theo thứ tự chữ cái hay không, nếu có, ta nói $A > B$

khi mà cả tên sản phẩm cũng giống nhau thì ta xét xem ngày-tháng-năm giao dịch của hai dòng thì thời gian của A có xếp sau B hay không, nếu có, ta nói $A > B$

khi mà cả ngày tháng năm giao dịch cũng giống nhau thì ta xét xem xâu tạo thành bởi tên người bán (seller), dấu gạch ngang và người mua (buyer) ở hai nút A, B, nếu xâu của A đứng sau xâu của B trong theo thứ tự chữ cái thì ta nói $A > B$. Chẳng hạn xâu này của A là "tom_jack" và xâu của B là "anna_bob" thì rõ ràng "tom_jack" > "anna_bob"

Nếu đến cả xâu của A và B cũng vẫn giống nhau thì ta không tạo ra nút trên cây đại diện cho B mà chỉ đơn giản gom thông tin của nút B vào để tính gộp cho nút A. Chẳng hạn số tiền, số sản phẩm của A lần lượt là 300 và 15 còn số tiền, số sản phẩm của B là 400 và 20 thì sau khi tính gộp, số tiền và số sản phẩm của A là 700 và 35. Các thông tin khác về ngày-tháng-năm, mã sản phẩm, tên seller và tên buyer vẫn được giữ nguyên.

a) Với cách so sánh các nút như ở trên, hãy xây dựng hàm `buildTree(int N)` nhận tham số là số lượng dòng dữ liệu hợp lệ và trả về con trỏ (mà trỏ đến nút gốc của cây). Hàm này trước khi kết thúc in ra số lượng nút trong cây **(1.5 điểm)**

b) hãy xây dựng hàm `height(T* root)` nhận đầu vào là con trỏ T (mà trỏ đến nút gốc của cây) và hàm `height` trả về độ cao của cây. (1.5 điểm)

c) Hãy cài đặt trong hàm `main` sao cho sau khi người dùng nhập xong ký tự \$ thì nó yêu cầu người dùng nhập vào giá trị N. Sau khi nhập xong N thì hai hàm trên được gọi và in ra số lượng nút trong cây cũng như in ra độ cao của cây.

Bài 3 (2 điểm)

Tương tự với input như ở bài 2, cài đặt các chức năng sau:

a) Kiểm tra xem ngày nào là ngày có nhiều giao dịch nhất giữa mọi người **(1 điểm)**

b) Yêu cầu người dùng nhập vào ngày tháng năm DD-MM-YYYY nào đó, cũng như tên người bán X và người mua Y, chương trình kiểm tra xem trong ngày tháng năm đó, X đã bán bao nhiêu sản phẩm cho B, tổng số tiền đã giao dịch **(1 điểm)**

Để đơn giản, không cần kiểm tra sự hợp lệ của ngày tháng năm

Yêu cầu:

Bài cuối sẽ cần phải cài đặt bảng băm để ra kết quả mau chóng khi mà bộ dữ liệu test có thể lên đến hàng triệu dòng.