
Câu 1 : (2 điểm)

Cho q là một hàng đợi với các hàm: **isEmpty(q)** để kiểm tra q có trống hay không, **dequeue(q)** để lấy phần tử đầu tiên của hàng đợi; **enqueue(q, k)** để thêm k vào cuối hàng đợi.

Cho đoạn chương trình sau: int i = 0; int k; while (!isEmpty(q)) { k = dequeue(q); cout << k << endl; if (i) enqueue(q, k); if (i == 0) i = 1 ; else i = 0; }	Cho hàng đợi q đang chứa các số nguyên theo thứ tự từ trái qua phải: 2, 5, 7, 1, 8. Anh / chị cho biết kết quả in ra màn hình của đoạn chương trình trên ?
---	---

Đáp án : 2 5 7 1 8 5 1 5 5 hoặc 8 1 7 5 2 1 5 1 1

Thang điểm :

- Kết quả hiển thị đúng được từng số (theo thứ tự đúng với đáp án) : 0,25 điểm / 1 số đúng
- Đúng hết đáp án (tất cả các số) : trọn vẹn điểm

Câu 2 : (2 điểm)

Cho một danh sách liên kết, mỗi phần tử lưu thông tin về một cửa hàng tiện lợi bao gồm các trường dữ liệu: *tên cửa hàng* (chuỗi ký tự); *số mặt hàng bán của cửa hàng* (số nguyên); *tỷ lệ đánh giá mức độ tốt* của khách hàng (số nguyên), *khoảng cách (km) đến cửa hàng* (số thực). Anh / chị hãy :

a. Định nghĩa cấu trúc dữ liệu lưu thông tin cửa hàng như mô tả trên. (1 điểm)

Đáp án câu 2.a :

- SV có thể sử dụng DSLK đơn hoặc kép (đôi) để định nghĩa CTDL

Thang điểm :

- SV khai báo đúng CTDL cửa hàng: 0,5 điểm
- SV khai báo đúng cấu trúc DSLK cửa hàng : 0,5 điểm
- SV làm đúng trọn vẹn yêu cầu : trọn vẹn điểm

b. Viết hàm nhập vào 10 cửa hàng **bằng giải thuật thêm vào cuối danh sách**. (1 điểm)

Đáp án câu 2.b:

- SV chỉ cần viết hàm theo đúng yêu cầu, không cần viết chương trình.
- SV sử dụng đúng giải thuật thêm vào cuối danh sách
- SV có thể sử dụng lại (hoặc không sử dụng lại) các hàm trung gian phổ biến của DSLK : **tạo danh sách rỗng, tạo phần tử (node) khi viết hàm nhập.**

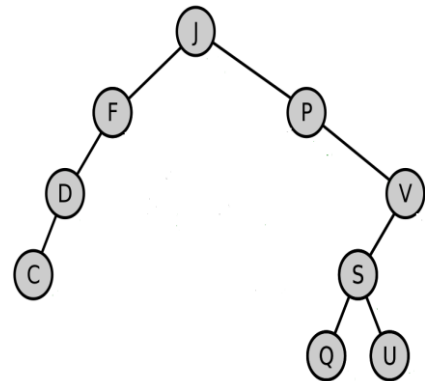
Thang điểm :

- SV viết hàm nhập vào 10 cửa hàng (phần tử) đúng, **nhưng không sử dụng đúng giải thuật thêm vào cuối danh sách** như yêu cầu : trừ 0,5 điểm.
- SV viết hàm nhập cửa hàng đúng, sử dụng đúng giải thuật thêm vào cuối danh sách như yêu cầu, **nhưng chỉ thêm vào được 1 cửa hàng (phần tử)** : trừ 0,5 điểm.
- SV viết hàm nhập đảm bảo đúng theo yêu cầu : trọn vẹn điểm.

Câu 3: (4 điểm)

Cho cây nhị phân tìm kiếm như hình, anh / chị hãy :

- Cho biết kết quả duyệt cây theo Left-Node-Right, Node-Left-Right. (1 điểm)
- Hãy cân bằng cây: cho biết nút bị mất cân bằng, loại mất cân bằng và vẽ cây khi cân bằng từng nút mất cân bằng. (1.5 điểm)
- Vẽ cây sau khi xóa nút F, S, J trên cây, cân bằng lại cây nếu cây bị mất cân bằng. (1.5 điểm).



Đáp án câu 3.a :

Câu 3.a :

- SV cần cho biết chính xác kết quả duyệt cây theo LNR, NLR

Thang điểm :

- SV có kết quả duyệt cây đúng LNR : 0.5 điểm
- SV có kết quả duyệt cây đúng NLR : 0.5 điểm
- SV có kết quả chỉ sai một nút trong kết quả duyệt cây NLR hoặc LNR : không có điểm của duyệt cây NLR hoặc LNR.

Đáp án câu 3.b:

Câu 3.b : SV có thể thực hiện theo 2 cách

- Cách 3.b.1 : SV có thể áp dụng các qui tắc của cây nhị phân tìm kiếm cân bằng để xác định nút bị mất cân bằng và tiến hành cân bằng cây

Thang điểm :

- Nhận diện nút mcb F, loại mcb LL, cân bằng cây : 0.5 điểm
- Nhận diện nút mcb V, loại mcb LL / LR, cân bằng cây : 0.5 điểm
- Nhận diện nút mcb P, loại mcb LL, cân bằng cây : 0.5 điểm

- Cách 3.b.2 : SV có thể duyệt cây để lấy ra lần lượt các node trên cây và tiến hành xây cây nhị phân tìm kiếm cân bằng, trong quá trình tạo cây nhị phân tìm kiếm cân bằng, nếu xảy ra mất cân bằng sẽ cân bằng lại.

Thang điểm :

- Thao tác duyệt cây để lấy ra lần lượt các node trên cây chính xác : 0.25 điểm
- Thao tác xây cây nhị phân tìm kiếm cân bằng và cân bằng (nếu có) chính xác: 1.25 điểm

Đáp án câu 3.c :

Câu 3.c: SV có thể thực hiện theo 2 cách

- Cách 3.c.1 : SV có thể áp dụng các qui tắc của cây nhị phân tìm kiếm cân bằng để xóa nút F, S, J trên kết quả từ câu 3.b (theo cách 3.b.1)

Thang điểm :

- Xóa nút F : 0.5 điểm
- Xóa nút S : 0.5 điểm
- Xóa nút J : 0.5 điểm

- Cách 3.c.2 : SV có thể xóa nút F,S,J trên cây từ câu 3.b (theo cách 3.b.2)

Thang điểm :

- Xóa nút F : 0.5 điểm
- Xóa nút S : 0.5 điểm
- Xóa nút J : 0.5 điểm

Câu 4 : (2 điểm)

Cho tập khóa $\mathbf{K} = \{76, 93, 40, 47, 10, 55, 24, 12, 18, 59, 28\}$ và hàm băm :

$$\mathbf{H}(\mathbf{key}) = \mathbf{key} \% 11 \text{ (key: khóa cần băm)}.$$

Anh / chị hãy thực hiện vẽ hình từng bước lưu trữ từng khóa trong \mathbf{K} vào bảng băm có kích thước $M=11$, dùng hàm băm \mathbf{H} và phương pháp **dò tuyến tính** để xử lý xung đột.

Đáp án:

- *SV có thể thêm các số lần lượt trong tập K theo thứ tự tùy ý trước sau, nếu xảy ra đụng độ xử lý bằng phương pháp dò tuyến tính.*
- *Vẽ hình từng bước khi thêm lần lượt các số để kiểm tra SV hiểu phương pháp.*

Thang điểm :

- *Vẽ bảng băm với kích thước 11 phần tử: 0,25*
- *Đánh chỉ số cho từng phần tử (từ 0 đến 10): 0,125*
- *Thêm số đầu tiên vào bảng băm: 0,25 (thể hiện SV biết dùng hàm băm để xác định địa chỉ)*
- *Thêm số vào bảng băm, trường hợp lần đầu tiên xảy ra đụng độ: 0,25 (thể hiện SV biết cách giải quyết đụng độ)*
- *Thêm 9 số còn lại, mỗi số 0,125*

HẾT