CSD201 – Assignment 02 – 90 phút

Đề bài gồm 2 phần

* Sinh viên có thể viết code trực tiếp vào các vị trí để trống, hoặc viết thêm các hàm phụ trợ và gọi những hàm này tại các vị trí thích hợp
* **Lưu ý: dùng f.writeBytes(… + "\n")** trong hàm ftraverse() để xuất dữ liệu ra file.

Phần 1. Stack (LIFO) và Queue(FIFO) (6 câu, mỗi câu 1 điểm, thời gian 45 phút)

1. Q1 – push: hàm thêm dữ liệu vào vị trí đầu tiên (vị trí đỉnh) của stack – tương tự như hàm addFirst của linked list.
2. Q1 – f2 – pop : hàm lấy dữ liệu của vị trí đầu tiên (vị trí đỉnh) của stack, sau đó xuất dữ liệu ra file, và xoá giá trị này khỏi stack – tương tự như hàm showFirst + removeFirst của linked list.
3. Q1 – f3 – peek : hàm lấy dữ liệu của vị trí đầu tiên (vị trí đỉnh) của stack, sau đó xuất dữ liệu ra file, tương tự như hàm showFirst của linked list.
4. Q2 – enQueue: hàm thêm dữ liệu vào vị trí cuối cùng của queue – tương tự như hàm addLast của linked list.
5. Q2 – f2 – deQueue : hàm lấy dữ liệu của vị trí đầu tiên của queue, sau đó xuất dữ liệu ra file và xoá giá trị này khỏi queue – tương tự như hàm showFirst + removeFirst của linked list.
6. Q2 – f3 – front : hàm lấy dữ liệu của vị trí đầu tiên của queue, sau đó xuất dữ liệu ra file, tương tự như hàm showFirst của linked list.

Phần 2. Priority Queue (4 câu, mỗi câu 1 điểm, thời gian 45 phút)

Class Patient có thông tin emergency level (eLevel), nếu eLevel>5 thì bệnh nhân đang trong tình trạng khẩn cấp, còn lại thì là thông thường (không khẩn cấp).

Class MyQueue có 2 head và 2 tail, ứng với hai hàng đợi cho bệnh nhân khẩn cấp (ehead, etail) và bệnh nhân thông thường (head, tail). Mỗi hàng đợi được vận hành theo nguyên tắc **First In First Out**.

**Quy tắc ưu tiên:** khám chữa bệnh cho bệnh nhân khẩn cấp trước, khi không còn ai là bệnh nhân khẩn cấp nữa, thì mới khám chữa bệnh cho bệnh nhân thông thường.

**\*\*\* Chú ý: xem kỹ đáp án để viết các dòng code xuất kết quả cho đúng**

VD: **“Emergency:3”** sẽ khác so với **“Emergency:3.0”,** cũng khác so với **“Emergency: 3”**

1. enQueue(): bệnh nhân nào thì được đưa vào hàng đợi tương ứng;
   1. Bệnh nhân khẩn cấp: quản lý bởi cặp con trỏ ehead và etail
   2. Bệnh nhân thông thường: quản lý bởi head và tail

Kết quả dự kiến trong file f1.txt có nội dung như sau:

Emergency: NVA-8.0 NVD-9.0 LVX-10.0

Normal: LVB-3.0 TTC-5.0 LTH-2.0 LHD-1.0 LMS-4.0

1. f2(): Đếm số lượng bệnh nhân ở hai dạng: khẩn cấp và thông thường. Kết quả trong file f2.txt có nội dung như sau:

Emergency:3

Normal:5

Emergency: NVA-8.0 NVD-9.0 LVX-10.0

Normal: LVB-3.0 TTC-5.0 LTH-2.0 LHD-1.0 LMS-4.0

1. f3(): deQueue theo quy tắc ưu tiên bên trên. Kết quả dự kiến trong file f3.txt có nội dung như sau:

NVA-8.0

Emergency: NVD-9.0 LVX-10.0

Normal: LVB-3.0 TTC-5.0 LTH-2.0 LHD-1.0 LMS-4.0

1. f4(): deQueueForAll – khám chữa bệnh cho tất cả bệnh nhân đang chờ. Kết quả dự kiến trong file f4.txt có nội dung như sau:

Patient (Emergency): NVA-8.0

Patient (Emergency): NVD-9.0

Patient (Emergency): LVX-10.0

Patient: LVB-3.0

Patient: TTC-5.0

Patient: LTH-2.0

Patient: LHD-1.0

Patient: LMS-4.0

Emergency:

Normal:

Lưu ý chung:

* SV được trao đổi nhưng không được làm ồn
* Được “tham khảo” code của bạn bên cạnh, nhưng không nên chép lại y chang
* Có thể search và tham khảo từ nhiều nguồn, nhưng nên **tự làm, tự gõ lại code**
* Lưu ý: các điều kiện ràng buộc của đề bài (price, ID, …)
* Lưu ý: cập nhật đầy đủ giá trị cho **head, tail, next, pre, size, …**
* Lưu ý: xét đủ các trường hợp đặc biệt như list rỗng, list có 1 phần tử, …
* Khi **chấm bài**, thầy **sẽ đổi data trong file data.txt**, do đó các em cần code đúng theo yêu cầu, tránh bị lệ thuộc quá nhiều vào data kèm theo đề.

Nộp bài:

* (a) Trước khi nộp bài: với cả 3 câu (Q1, Q2, Q3), các em cần thực hiện:
  + **clean and build** project,
  + sau đó **đổi tên** thư mục **dist** thành **run**.
* (b) Nén folder Q1, Q2 và Q3 thành file nén **Code\_A02.zip** (chọn 3 folder Q1, Q2 và Q3, click chuột phải, sau đó chọn Add to archive …)
* (c) Sau đó nộp file nén lên LMS theo đúng deadline

----- Hết -----