BÀI TẬP CHƯƠNG 2

Bài 1:

Danh sách đặc là một danh sách các phần tử trong danh sách có cùng kiểu dữ liệu, và được cấp phát liên tục trong bộ nhớ

Ví dụ: Danh sách điểm của học sinh Toán:

* Kiểu dữ liệu là double
* Cấp phát liên tục cho n học sinh

Bài 2:

* Một Danh sách liên kết (Linked List) là một dãy các cấu trúc dữ liệu được kết nối với nhau thông qua các liên kết (link). Hiểu một cách đơn giản thì Danh sách liên kết là một cấu trúc dữ liệu bao gồm một nhóm các nút (node) tạo thành một chuỗi. Mỗi nút gồm dữ liệu ở nút đó và tham chiếu đến nút kế tiếp trong chuỗi.
* Danh sách liên kết theo chương trình học có 2 loại:

.Danh sách liên kết đơn: là một danh sách mà các phần tử được cấp phát rời rạc nhau và được cố định trong bộ nhớ. Mỗi phần tử trong danh sách gồm có 2 thành phần:

+Phần 1: vùng thông tin chứa giá trị cần quản lý

+Phần 2: vùng liên kết chứa địa chỉ bộ nhớ của phần tử kế tiếp.

.Danh sách liên kết kép:là một danh sách liên kết mà mỗi phần tử trong danh sách bao gồm 3 Thành phần:

+ vùng chứa thông tin.

+ Vùng liên kết(next) trỏ đến phần tử đứng liền sau nó.

+Vùng liên kết (previous) trỏ đến phần tử đứng liền trước nó.

* Ví dụ: Chèn vào một danh sách sau phần tử q

+ Chèn vào danh sách sau một phần tử q nào đó, chèn vào giữa danh sách không cần cập nhập lại hai con trỏ pHead và pTail tuy nhiên chúng ta cần hết sức cần thận để tránh mất dữ liệu phía sau.

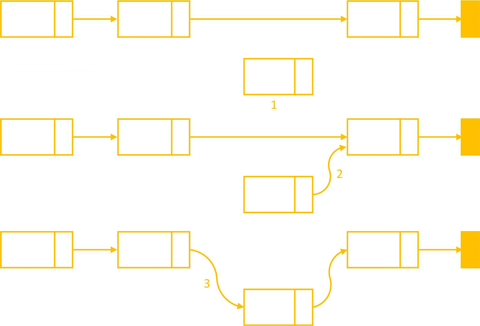
**Bước 1:** cấp phát một Node mới - new\_element.

**Bước 2:** gán pNext của Node mới bằng địa chỉ của Node sau q.

1. new\_element ->pNext = q-> pNext;

**Bước 3:** cập nhập lại pNext của Node q trỏ đến Node mới.

1. q-> pNext = new\_element;



Bài 3:

* Tại sao nói STACK và QUEUE là danh sách hạn chế?

Vì đều lấy ra, them vào có quy tắc.

* Ví dụ:

+STACK: Chuyển đổi số thập phân sang nhị phân: Khi chuyển đổi số thập phân sang nhị phân, chúng ta sẽ thưc hiện thao tác chia số thập phân này cho 2 và viết phần dư ngược lại thứ tự mà nó được sinh ra. Sử dụng stack, ta lưu lần lượt phần dư sau mỗi lần chia và sau khi kết thúc thao tác chia, đọc stack vừa rồi sẽ cho ra biểu diễn nhị phân cần tìm.

+QUEUE:

* Xử lý các lệnh trong máy tính: Điều này khá là dễ hiểu nên mình không bình luận thêm.
* Kiểm tra chuỗi Palindrome: Một chuỗi được gọi là có tính chất Palindrome nếu nó có tính chât đối xứng, tức là viết xuôi cũng giống viết ngược, ví dụ như "aaAaa". Để kiểm tra tính chất này của một chuỗi bất kì, ta đọc chuỗi bởi 2 cấu trúc riêng biệt là stack và queue. Sau đó, lấy ra từng phần tử trong stack và queue để so sánh với nhau. Nếu tất cả các phần tử trong stack đều giống với phần tử trong queue ở vị trí tương ứng thì chuỗi đó có tính chất Palindrome.

Bài 4:

* **FIFO (First in First out –**Vào trước Ra trước): Với phương pháp này, các lô hàng đầu tiên của hàng hoá nhập vào nhà kho sẽ là hàng hoá đầu tiên được xuất ra khỏi kho – từ đó được gửi vào các cửa hàng hoặc gửi trực tiếp đến khách hàng.
* **LIFO (Last in First Out – Vào sau Ra trước):**Ở phương pháp này, các hàng hoá được nhập vào kho gần nhất sẽ được xuất ra đầu tiên. Hàng hoá mới được sử dụng trước, dùng ưu tiên hơn hàng hoá cũ.
* **Ví dụ:**
* FIFO: Các mặt hàng như sữa. Khi bạn lấy sữa từ kho hàng của bạn và bán nó trong cửa hàng, bạn muốn các lô hàng đầu tiên ở phía trước kệ hàng. Nếu không, khách hàng sẽ mua lô mới trước, các lô hàng cũ sẽ bị quá hạn.
* LIFO: các sản phẩm gần đây nhất của bạn có chi phí sản xuất tăng. Nếu bạn bán những sản phẩm đó với giá theo chi phí của các sản phẩm được sản xuất trong quá khứ, bạn sẽ thấy chi phí sản xuất ở mức thấp và lợi nhuận thu được ở mức cao bởi vì bạn đang làm việc với các thông tin cũ khác hẳn so với thông tin thực tế tại thời điểm hiện tại.

Bài 6:

Danh sách liên kết có thể xử lý những vấn đề trong máy tính:

* Cung cấp giải pháp để chứa cấu trúc dữ liệu tuyến tính.
* Dễ dàng thêm hoặc xóa các phần tử trong danh sách mà không cần phải cấp phát hoặc tổ chức lại trật tự của mảng.
* Cấp phát bộ nhớ động

Bài 7:

Cấu trúc dữ liệu động là cấu trúc được cấp phát rời rạc, không giới hạn

Ví dụ:

int\*    p1, p2;

// cấp phát vùng nhớ cho 1 biến động kiểu int

p1 = (int\*)malloc(sizeof(int));

p1\* = 5;       // đặt giá trị 5 cho biến động p1

// cấp phát biến động kiểu mảng gồm 10 phần tử kiểu int

p2 = (int\*)calloc(10, sizeof(int));

(p2+3)\* = 0; // đặt giá trị 0 cho phần tử thứ 4                                   // của mảng p2

free(p1); free(p2);