

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

HOMEWORK 5.1 – Danh sách liên kết

---oOo---

1. Khai báo DSLK đơn như sau:

```
struct NODE
{
    int      Key;
    NODE*    pNode;
};
```

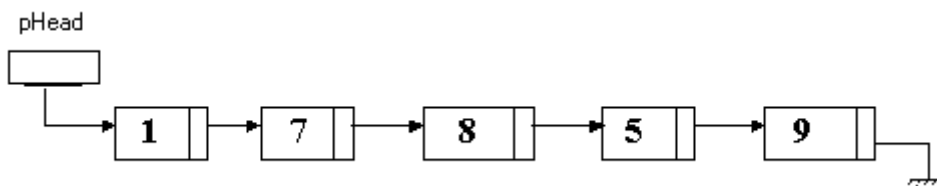
Trong đó, NODE* pNode có nghĩa là:

- Con trỏ tới vùng dữ liệu của phần tử hiện tại.
- Con trỏ tới phần tử kế tiếp trong DSLK.
- Con trỏ tới địa chỉ vùng nhớ của phần tử trước đó trong DSLK.
- Con trỏ tới địa chỉ vùng nhớ đầu tiên trong DSLK.

2. Đặc điểm nào sau đây không phải là của danh sách liên kết

- Sử dụng con trỏ (pointer)
- Các node lưu trữ liên tiếp nhau trong bộ nhớ
- Có thể mở rộng tùy ý (chỉ giới hạn bởi dung lượng bộ nhớ)
- Thao tác Thêm/Xóa không cần phải dịch chuyển phần tử

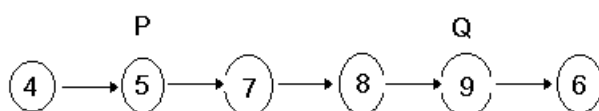
3. Cho một danh sách liên kết đơn như sau:



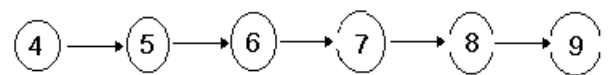
- p1 = pHead->pNext;
 - p3 = p1->pNext->pNext;
 - p3->data = p1->data;
 - pCurr = p1;
- ```
while (pCurr != NULL) {
 pCurr->data *= 5;
 pCurr = pCurr->pNext;
}
```

Hãy thực hiện tuần tự các lệnh từ [a] đến [d], vẽ lại DSLK sau mỗi lệnh để thể hiện rõ sự thay đổi của DSLK.

#### 4. Cho một DSLK đơn như hình vẽ 1. Cho trước con trỏ P, Q tại các vị trí tương ứng



Hình 1



Hình 2

Hãy viết các lệnh cần thiết để chuyển DSLK thành dạng biểu diễn như hình 2.

**5. Cho một DSLK với phần tử đầu là pHead và hàm fn như sau.**

Hãy cho biết kết quả khi gọi hàm fn(pHead).

```
int fn(NODE *p)
{
 if (p!=NULL) return (1 + fn(p->Next));
 else return 0;
}
```

**6. Cho một DSLK đơn có phần tử đầu là pHead, mỗi phần tử trong DSLK có các field:**

Key: khóa, kiểu int

pNext: con trỏ đến phần tử kế tiếp trong DSLK

- Hãy viết một hàm không đệ qui đếm số phần tử trong DSLK.
- Hãy viết một hàm đệ qui đếm số phần tử trong DSLK.

**7. Hãy viết hàm “Xóa phần tử đầu tiên trong một danh sách liên kết đơn”**

Hàm có dạng như sau:

```
NODE * Delete_1st_item_in_list(NODE *pHead);
```

với pHead là con trỏ đầu danh sách. Yêu cầu xử lý được cho các trường hợp:

- Danh sách rỗng
- Danh sách chỉ có 1 phần tử
- Danh sách có nhiều phần tử
- Nếu danh sách rỗng hoặc danh sách chỉ có 1 phần tử thì sau khi xóa, hàm trả về giá trị NULL
- Nếu danh sách có nhiều phần tử thì sau khi xóa, hàm trả về con trỏ của phần tử kế sau pHead

**8. Cho 1 danh sách liên kết đơn trong đó, mỗi phần tử chứa 1 số nguyên.**

Hãy viết hàm xóa phần tử có giá trị lớn nhất trong danh sách.

Lưu ý:

- Chỉ được phép duyệt danh sách 1 lần
- Nếu có nhiều phần tử lớn nhất giống nhau thì xóa phần tử đầu tiên

**9. Giả sử ta cần lưu trữ một danh sách gồm N sinh viên (N chưa biết trước).**

Mỗi sinh viên có các thuộc tính dữ liệu sau:

- Mã SV: chuỗi 8 ký tự
- Họ tên: chuỗi 50 ký tự
- Ngày sinh: dd/mm/yyyy
- Điểm trung bình: số thực
- Giới tính: Nam/Nữ

Yêu cầu:

- Hãy trình bày 2 cấu trúc dữ liệu khác nhau để lưu trữ danh sách nêu trên. Với mỗi cấu trúc dữ liệu, hãy:
  - Khai báo cấu trúc đó bằng C/C++.
  - Cho biết cần bao nhiêu byte để lưu trữ N phần tử?
- Lập bảng so sánh (ngắn gọn, tối đa 3 chỉ tiêu) hai cấu trúc dữ liệu trên theo mẫu sau:

| Chỉ tiêu so sánh | Cấu trúc dữ liệu 1 | Cấu trúc dữ liệu 2 |
|------------------|--------------------|--------------------|
|                  |                    |                    |
|                  |                    |                    |
|                  |                    |                    |