Bài tập Tuần 4-Con trỏ, DSLK, Cây Nhị Phân

Câu 1 – DSLK

Thông tin về một bộ phim được định nghĩa theo cấu trúc sau:

```
struct Phim
{
    string  Ten;            //tên phim
    string  NuocSanXuat;    //nước sản xuất
    int     NamSanXuat;     //năm sản xuất
    string  TheLoai;        //thể loại phim
    float   DiemDanhGia;    //điểm đánh giá
    Phim*   pNext;          //con trỏ đến thông tin của phim tiếp theo
};
```

a. Giả sử danh sách phim đã được sắp xếp tăng dần theo năm sản xuất và được quản lý bởi con trỏ pHead trỏ tới phần tử đầu tiên của danh sách, hãy viết hàm thêm một bộ phim mới vào danh sách phim sao cho vẫn đảm bảo tính chất tăng dần theo năm sản xuất. Hàm thêm phim mới được mô tả như dưới đây:
   Phim* ThemPhim(Phim*& pHead, string Ten, string NuocSanXuat
                  int NamSanXuat, string TheLoai, float DiemDanhGia);
   Ghi chú: các tham số Ten, NuocSanXuat, NamSanXuat, TheLoai và DiemDanhGia là thông tin của bộ phim cần bổ sung vào danh sách. Kết quả trả về của hàm là con trỏ đến bộ phim vừa được bổ sung vào danh sách.

b. Viết hàm các phim có thể loại phim là strTheLoai và điểm đánh giá tối thiểu là minDiemDanhGia theo mô tả hàm dưới đây:
   Phim* TimKiemPhim(Phim* pHead, string strTheLoai, float minDiemDanhGia);
   Ghi chú:
   - Kết quả trả về của hàm TimKiemPhim là con trỏ đến bộ phim đầu tiên trong danh sách các bộ phim tìm được hoặc con trỏ NULL nếu không tìm được bộ phim nào thoả điều kiện.

Câu 2 – Cây nhị phân

Thông tin về một nút trong cây nhị phân tìm kiếm (Binary Search Tree) được định nghĩa theo cấu trúc sau:

```
struct NODE{
    int     value;      //giá trị của nút
    NODE*   pLeft;      //nút con bên trái, nếu có
    NODE*   pRight;     //nút con bên phải, nếu có
};
```

a. Viết hàm chèn một giá trị x vào cây nhị phân tìm kiếm có gốc là nút root theo mô tả hàm dưới đây:
   NODE* InsertNode(NODE*& root, int x);

Ghi chú: giá trị trả về của hàm chính là con trỏ đến nút chứa giá trị x vừa thêm vào cây. Nếu trong cây đã có sẵn nút chứa giá trị x, trả về nút này.

b. Viết hàm đếm số lượng nút trên cây (có nút gốc là root) có giá trị là số chẵn theo mô tả hàm sau đây:
   int CountEvenNodes(NODE* root)

c. Viết hàm xác định nút có giá trị lớn nhất không vượt quá giá trị x cho trước trên cây (có nút gốc là root) theo mô tả hàm sau đây:

NODE* FindNode(NODE* root, int x);

Nếu không tìm được nút nào thoả điều kiện, trả về NULL;

Câu 3.

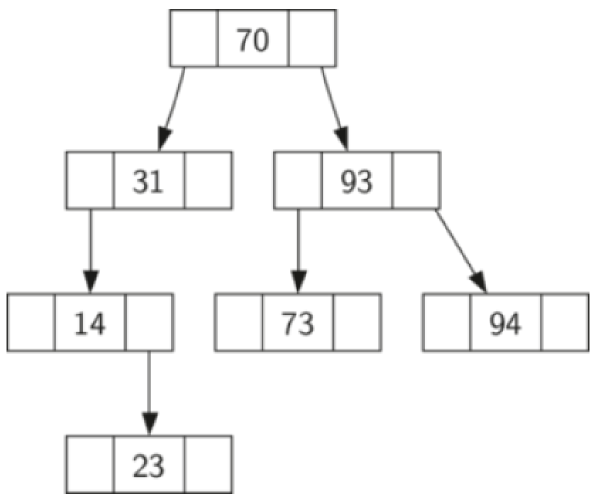
Thông tin về một nút trong cây nhị phân tìm kiếm (Binary Search Tree) được định nghĩa theo cấu trúc sau:

```
struct NODE{
	int		value;		//giá trị của nút
	NODE*	pLeft;		//nút con bên trái, nếu có
	NODE*	pRight;		//nút con bên phải, nếu có
};
```

a. Đếm các nút ở mức h.
b. Duyệt cây (BST) theo mức.
c. Tính tổng các nút trong cây có giá trị >=x.

Câu 4. Viết chương trình cài đặt cây nhị phân và kiểm thử lại các hàm được viết ở Câu 2 và Câu 3 ở trên.

Với một minh hoạ của cây nhị phân sau:



================== ***** ==================