

Câu 1 (1.5 điểm). Tính số phép toán cơ bản nhiều nhất mà thuật toán dưới đây cần phải thực hiện. Xác định độ phức tạp tiệm cận của thuật toán.

```
int Count2(int *a, int n)
{
    int k=0;
    for(int i=1; i<=n; i++)
        for(int j=i+1; j<=n; j++)
            if((a[i]+a[j])%2==0 && a[i]>a[j])
                k++;
    return k;
}
```

Câu 2. (2.0 điểm) Thực hiện các yêu cầu sau đây:

- Trình bày về cấu trúc dữ liệu danh sách liên kết đơn: Khái niệm, các thao tác, vẽ hình mô tả
- Khi xây dựng lớp mẫu Node có dữ liệu gồm **public: T elem; Node *next;** và lớp mẫu Slist có dữ liệu gồm **int n; Node<T> *Head, *Trail;** hãy lập trình các phương thức sau trong lớp Slist: (không sử dụng bộ lặp)
 - Hàm **void Slist<T>::erase(T x)** xoá một phần tử x (nếu có) ra khỏi Slist
 - Hàm **void Slist<T>::push_front(T x)** thêm phần tử vào đầu Slist.

Câu 3. (3.0 điểm) Thực hiện các yêu cầu sau đây:

- Xây dựng lớp biểu diễn các đối tượng **Tivi** gồm các thuộc tính: mã hàng (không trùng nhau), Hãng SX, tên Tivi, đơn giá, số lượng; Các toán tử nhập >>, xuất <<, phương thức lấy mã máy.
- Xây dựng lớp quản lý một danh sách Tivi bằng **vector** (sử dụng lớp vector của C++), có các phương thức sau: Thêm 1 Tivi, Xóa một Tivi theo mã, Tính tổng giá trị hàng hoá tồn kho theo 1 hãng sản xuất (nếu không có trả về 0). In danh sách Tivi lên màn hình, Tìm kiếm Tivi theo mã bằng thuật toán tìm kiếm nhị phân (giả sử danh sách Tivi đã được sắp xếp tăng dần theo mã).

Câu 4. (2.0 điểm) Thực hiện các yêu cầu sau đây:

- Trình bày ý tưởng của thuật toán sắp xếp Quicksort
- Lập trình thuật toán Quicksort

Câu 5 (1.5 điểm).

- Mô tả quá trình phân hoạch dãy số sau đây bằng thuật toán **Partition** trong thuật toán Quicksort.

86, 74, 38, 45, 24, 129, 69, 154, 35, 75, 12, 543, 153.

- Vẽ cây tìm kiếm nhị phân với các nút lần lượt có giá trị sau: 299, 125, 135, 555, 66, 333, 169, 567, 80, 190, 226, 78.