

**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, \*Tail;** hãy lập trình các phương thức trong lớp Slist.

**Câu 2. (3.5 điểm) Thực hiện các yêu cầu sau đây**

- a. Xây dựng lớp biểu diễn các đối tượng nhân viên của một công ty gồm các thuộc tính: mã, họ tên, năm sinh, giới tính, lương; các toán tử nhập `>>`, xuất `<<`, phương thức lấy mã.
- b. Xây dựng lớp **quản lý** một danh sách nhân viên của một công ty bằng **danh sách liên kết đơn** (giả sử các lớp tạo danh sách liên kết đơn đã có), có các phương thức sau:
  - Tạo 1 danh sách có  $n$  nhân viên, các nhân viên lần lượt được thêm vào cuối danh sách.
  - Xóa một nhân viên có mã là  $x$ ,  $x$  được nhập từ bàn phím.
  - In danh sách các nhân viên có lương lớn hơn 2.000.000 lên màn hình.

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

- a. Xây dựng một lớp quản lý một dãy số (DaySo) bất kỳ có các phương thức sau:
  - Nhập vào một dãy n số
  - Sắp xếp dãy số bằng thuật toán sắp xếp **Quicksort**
  - Tìm kiếm phần tử của dãy bằng thuật toán tìm **kiếm nhị phân ở trên mảng**
  - In dãy số lên màn hình
- b. Viết hàm **main** tạo một đối tượng của lớp **DaySo**, lần lượt gọi các phương thức của đối tượng này thực hiện.
- c. Mô tả quá trình tạo mảng biểu diễn cây heap của dãy số sau đây: 23, 54, 1, 53, 76, 123, 43, 87, 321, 43.
- d. 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.