**Câu 1 (7 điểm)** Xây dựng class CPoint gồm

* Hai thuộc tính, không cho phép truy cập trực tiếp từ bên ngoài class là: Tung độ x và hoành độ y. **(1 điểm)**
* Và các phương thức sau:
  + Constructor không tham số, có tham số của CPoint để khởi tạo các thuộc tính. **(1 điểm)**
  + Nhập một điểm từ bàn phím. **(1 điểm)**
  + Xuất thông tin của một điểm ra màn hình. **(1 điểm)**
  + Di chuyển điểm sang phải 1 đơn vị. **(1 điểm)**
  + Kiểm tra 3 điểm thẳng hàng. (Giả sử a, b, c lần lượt là khoảng cách từ A->B, B->C, A->C. Điều kiện để 3 điểm thẳng hàng là a+b=c hoặc a+c=b hoặc b+c=a) **(2 điểm)**

**Câu 1 (3 điểm)** Chương trình chính để thực hiện các công việc sau:

* Tạo ra một object (P1) thuộc class CPoint được khởi tạo bởi các giá trị tùy ý và một object khác (P2) nhập từ bàn phím. **(1 điểm)**
* Tính khoảng cách giữa P1 và P2. Di chuyển P2 sang phải 1 đơn vị và xuất thông tin của P2 ra màn hình. **(1 điểm)**
* Tạo ra một object khác (P3), kiểm tra xem 3 điểm: P1, P2(sau khi đã di chuyển), P3 có thẳng hàng hay không. **(1 điểm)**

**(3 điểm)** Xây dựng lớp nhân viên:

Thuộc tính: Mã nhân viên, Họ tên, Tuổi

Phương thức: Nhập, Xuất, Kiểm tra về hưu (điều kiện về hưu: tuổi >=60)

**(3 điểm)** Xây dựng lớp công nhân kế thừa lớp nhân viên:

Thuộc tính: Mức lương, Ngày công, Tay nghề

Phương thức: Nhập, In, Tính lương (mức lương\* ngày công)

**(2 điểm)** Xây dựng lớp cán bộ kế thừa lớp nhân viên:

Thuộc tính: Hệ số lương, Phụ cấp, Trình độ

Phương thức: Nhập, In, Tính lương (hệ số lương\* 14500 + phụ cấp)

Xây dựng chương trình chính

* Tạo ra danh sách gồm n nhân viên (bao gồm cả công nhân và cán bộ) **(1 điểm)**
* In ra màn hình danh sách nhân viên đã về hưu vừa nhập cùng lương của nhân viên đó. **(0.5 điểm)**
* Sắp xếp danh sách nhân viên tăng dần về lương **(0.5 điểm)**

**(3 điểm)** Xây dựng lớp nhân viên:

Thuộc tính: Mã nhân viên, Họ tên, Tuổi

Phương thức: Nhập, Xuất, Kiểm tra về hưu (điều kiện về hưu: tuổi >=60)

**(3 điểm)** Xây dựng lớp công nhân kế thừa lớp nhân viên:

Thuộc tính: Mức lương, Ngày công, Tay nghề

Phương thức: Nhập, In, Tính lương (mức lương\* ngày công)

**(2 điểm)** Xây dựng lớp cán bộ kế thừa lớp nhân viên:

Thuộc tính: Hệ số lương, Phụ cấp, Trình độ

Phương thức: Nhập, In, Tính lương (hệ số lương\* 14500 + phụ cấp)

Xây dựng chương trình chính

* Tạo ra danh sách gồm n nhân viên (bao gồm cả công nhân và cán bộ) **(1 điểm)**
* In ra thông tin của của nhân viên chưa về hưu có tiền lương cao nhất **(0.5 điểm)**
* Sắp xếp danh sách nhân viên theo tên (tăng dần theo alphabet A-Z) **(0.5 điểm)**

**(4 điểm)** Xây dựng lớp ThoiGian gồm:

Thuộc tính: Gio, Phut, Giay

Phương thức:

- Khởi tạo.

- Nạp chồng toán tử: >>, <<

- Hàm bạn: KhoangCach (ThoiGian a, ThoiGian b): tính khoảng cách giữa

2 mốc ThoiGian.

- Nạp chồng toán tử lớn hơn: >

**(4 điểm)** Xây dựng một khuôn hình DanhSach gồm:

Thuộc tính: T \*arr; Size

Phương thức:

- Khởi tạo

- Nạp chồng toán tử: >>, <<

- Sắp xếp danh sách tăng dần theo thời gian

Chương trình chính:

- Nhập vào từ bàn phím DanhSach các mốc ThoiGian. **(1 điểm)**

- Sắp xếp danh sách theo chiều tăng dần. **(0.5 điểm)**

- Hiển thị danh sách sau sắp xếp tăng dần theo thời gian. **(0.5 điểm)**

Một đơn vị sản xuất gồm có các nhân viên là công nhân, kỹ sư

**(2.5 điểm)** Xây dựng các lớp NhanVien

* Mỗi nhân viên cần quản lý các thuộc tính: Họ tên, năm sinh, giới tính, địa chỉ
* Xây dựng các hàm khởi tạo, định nghĩa toán tử >>, << để nhập, hiển thị thông tin các thuộc tính của các lớp.

**(2.5 điểm)** Xây dựng các lớp CongNhan kế thừa từ lớp NhanVien

* Mỗi công nhân cần quản lý: tiền lương, công việc.
* Xây dựng các hàm khởi tạo, định nghĩa toán tử >>, << để nhập, hiển thị thông tin các thuộc tính của các lớp.

**(3 điểm)** Xây dựng các lớp KySu kế thừa từ lớp NhanVien

* Mỗi công nhân cần quản lý: Ngành đào tạo, Bậc lương (công nhân bậc 3/7, bậc 4/7 …).
* Xây dựng các hàm khởi tạo, định nghĩa toán tử >>, << để nhập, hiển thị thông tin các thuộc tính của các lớp.

**(2 điểm)** Xây dựng lớp QLNV cài đặt các phương thức thực hiện các chức sau:

* Nhập thông tin mới cho nhân viên (Hỏi người dùng muốn nhập cho: công nhân hay kỹ sư và nhập đúng thông tin cho đối tượng đó). **(1 điểm)**
* Hiển thị thông tin về danh sách các cán bộ. **(0.5 điểm)**
* Tìm kiếm nhân viên theo họ tên **(0.5 điểm)**

Xây dựng lớp PHANSO gồm:

* Thành phần dữ liệu: tu, mau. **(0.5 điểm)**
* Phương thức:
* Hàm khởi tạo không có tham số và có tham số. **(1.5 điểm)**
* In phân số ra màn hình. Lưu ý nếu tử chia hết cho mẫu thì không in ra dạng phân số mà in ra kết quả của phép chia tu/mau **(1.5 điểm)**
* Nhập phân số từ bàn phím. Lưu ý nếu mẫu bằng 0 thì phải nhập lại mẫu. **(1.5 điểm)**
* Tìm ước số chung lớn nhất của tử và mẫu **(1 điểm)**
* Rút gon phân số tu/mau về dạng tối giản **(1 điểm)**
* Định nghĩa các toán tử - (dùng hàm friend), \*(không dùng hàm friend) hai phân số. In kết quả lên màn hình. **(2 điểm)**

**(1 điểm)** Viết chương trình chính nhập vào hai phân số bất kỳ để kiểm tra các hàm vừa viết. In các kết quả lên màn hình

**(4 điểm)** Xây dựng lớp Oxy gồm:

Thuộc tính: float x, y;

Phương thức:

- Cấu tử.

- Hủy

- Nạp chồng toán tử nhập

- Nạp chồng toán tử xuất

- Nạp chồng toán tử - (tính khoảng cách giữa 2 Oxy)

**(4 điểm)** Xây dựng khuôn hình DanhSach gồm:

Thuộc tính: T \*ds; int Size

Phương thức:

- Cấu tử.

- Hủy

- Nạp chồng toán tử nhập (cấp phát động)

- Nạp chồng toán tử xuất

- Hàm MAX trả về khoảng cách xa nhất giữa 2 giá trị bất kỳ trong ds.

**(2 điểm)** Chương trình chính: Nhập vào từ bàn phím 1 DanhSach. Hiển thị danh sách; hiển thị khoảng cách xa nhất giữa 2 giá trị bất kỳ trong DanhSach đã nhập

Xây dựng lớp số phức (Complex) có dạng real+ image\*i:

* Các thuộc tính: real, image; **(1 điểm)**
* Các phương thức khởi tạo, hủy. **(2 điểm)**
* Định nghĩa các toán tử sau trên lớp Complex:

+ Toán tử cộng (+) **(1 điểm)**

+ Toán tử trừ dùng hàm friend (\*) **(1 điểm)**

+ Toán tử nhập (>>) **(1 điểm)**

+ Toán tử xuất (<<) **(1 điểm)**

Chương trình chính:

- Nhập vào 2 số phức a, b. **(0.5 điểm)**

- Tính tổng, hiệu 2 số phức a và b đã nhập, hiển thị kết quả ra màn hình. **(0.5 điểm)**

- Nhập vào 1 mảng các số phức, sắp xếp mảng vừa nhập theo thứ tự phần thực tăng

dần, hiển thị mảng đã nhập ra màn hình. **(2 điểm)**

Xây dựng lớp **CauThu** gồm:

Thuộc tính: HoTen, Tuoi, ViTri, CanNang, ChieuCao **(0.5 điểm)**

Phương thức:

- Khởi tạo, hủy **(1 điểm)**

- Nạp chồng toán tử >>, << **(1.5 điểm)**

- Nạp chồng toán tử < (Một **CauThu** la nhỏ hơn nếu chiều cao thấp hơn, nếu chiều cao bằng nhau thì xét cân nặng cao hơn) **(1 điểm)**

Xây dựng khuôn hình **DanhSach** gồm:

Thuộc tính: T \*ds; int Size **(0.5 điểm)**

Phương thức:

- Khởi tạo, hủy. **(1 điểm)**

- Nạp chồng toán tử >>, << **(1.5 điểm)**

- Sắp xếp DanhSach theo chiều giảm dần. **(1 điểm)**

Chương trình chính: Nhập vào từ bàn phìm:

- 1 danh sách n số nguyên **(0.5 điểm)**

- 1 danh sách m CauThu. **(0.5 điểm)**

- Sắp xếp 2 danh sách đã nhập, hiển thị thông tin lên màn hình **(1 điểm)**

Xây dựng lớp **Date** có các thành phần sau:

Thuộc tính: Ngay, Thang, Nam **(0.5 điểm)**

Phương thức:

- Khởi tạo. **(1 điểm)**

- Nạp chồng toán tử: >> (nhập), << (xuất) **(1.5 điểm)**

- Nạp chồng toán tử lớn hơn: > **(1 điểm)**

Xây dựng một khuôn hình lớp DanhSach gồm:

Thuộc tính: T \*arr; int Size **(0.5 điểm)**

Phương thức:

- Khởi tạo **(1 điểm)**

- Nạp chồng toán tử: >> (nhập), << (xuất) **(1.5 điểm)**

- Sắp xếp mảng danh sách theo chiều tăng dần **(1 điểm)**

Chương trình chính:

- Nhập vào từ bàn phím danh sách các **số nguyên**, sắp xếp danh sách theo chiều tăng dần và hiển thị danh sách sau sắp xếp. **(1 điểm)**

- Nhập vào từ bàn phím danh sách các **Date**, sắp xếp danh sách theo chiều tăng dần và hiển thị danh sách sau sắp xếp. **(1 điểm)**