

## LAB 8. KIỂU CẤU TRÚC

THỜI LƯỢNG: 6 TIẾT

### A. Mục tiêu

- Giúp sinh viên hiểu rõ và thực hiện thuần thục các kỹ thuật xử lý trên kiểu dữ liệu cấu trúc.
- Sau khi hoàn thành bài thực hành này, sinh viên cần phải:
  - Nắm vững các định nghĩa kiểu cấu trúc, biến cấu trúc.
  - Nắm vững các thao tác nhập - xuất giá trị cho các thành phần của cấu trúc.
  - Biết cách định nghĩa và sử dụng kiểu dữ liệu **mảng cấu trúc** bằng từ khóa **typedef**.
  - Vận dụng được các kỹ thuật tìm kiếm, sắp xếp vào mảng cấu trúc.
  - Hiểu rõ cơ chế gọi hàm và truyền tham số có kiểu cấu trúc.

### B. Yêu cầu

- Kết quả thực tập phần D (hướng dẫn thực hành) được thực hiện tại phòng Lab theo yêu cầu :
  - Thời gian thực hiện : 4 tiết
  - Tạo thư mục, đặt tên là **MSSV\_Lab08\_D\_HD**, để lưu bài làm. Trong đó, MSSV là mã số của sinh viên.
  - Các bài 1,2,3: tạo các project theo hướng dẫn, lưu trữ trong thư mục trên
  - Xóa thư mục Debug trong các project
  - Nén thư mục **MSSV\_Lab08\_D\_HD**
  - **Giáo viên thu bài qua mạng tại phòng lab vào cuối buổi thực tập thứ 13**
- Kết quả thực tập phần E (bài tập bắt buộc) được thực hiện tại phòng Lab theo yêu cầu :
  - Thời gian thực hiện : 2 tiết
  - Tạo thư mục, đặt tên là **MSSV\_Lab08\_E\_BB**, để lưu bài làm.
  - Chọn 1 bài trong các bài phần bắt buộc : tạo project lưu trữ trong thư mục trên, thực hiện theo yêu cầu.
  - Xóa thư mục Debug trong các project
  - Nén thư mục **MSSV\_Lab08\_E\_BB**
  - **Giáo viên thu bài qua mạng tại phòng lab vào cuối tiết 2 buổi thực tập thứ 14**

### C. Ôn tập lý thuyết

#### 1. Cách định nghĩa kiểu cấu trúc và khai báo biến cấu trúc

| Định nghĩa kiểu cấu trúc                                    |   |  |
|---|---|--|
| Dạng 1  | Dạng 2  | Dạng 3   |
| <pre>struct TênKiểuCấuTrúc {     // Các thành phần };</pre> | <pre>struct TênKiểuCấuTrúc {     // Các thành phần } danh_sách_các_biến_CT;</pre> | <pre>struct {     // Các thành phần } danh_sách_các_biến_CT;</pre> |
| Ví dụ   |   |  |
| <pre>struct NgayThang {     unsigned int Ngay;</pre>        | <pre>struct NgayThang {     unsigned int Ngay;</pre>                              | <pre>struct NgayThang {     unsigned int Ngay;</pre>               |

|   |   |  |
|---|---|--|
| <pre> unsigned int Thang; unsigned int Nam;  }; </pre>  | <pre> unsigned int Thang; unsigned int Nam;  }; </pre>  | <pre> unsigned int Thang; unsigned int Nam;  }; </pre>   |
| <pre> struct NhanVien {     unsigned int MaSo;     char HoTen[50];     char DiaChi[100];     NgayThang NgaySinh;     int SoDT; }; // Khai báo 2 biến cấu trúc NhanVien an, binh; </pre> | <pre> struct NhanVien {     unsigned int MaSo;     char HoTen[50];     char DiaChi[100];     NgayThang NgaySinh;     int SoDT; } an, binh; </pre> | <pre> struct {     unsigned int MaSo;     char HoTen[50];     char DiaChi[100];     NgayThang NgaySinh;     int SoDT; } an, binh; </pre> |

Trong đó:

- **TênKiểuCấuTrúc**: là tên kiểu dữ liệu cấu trúc, do lập trình viên tự đặt theo quy tắc đặt tên.
- Mỗi thành phần của cấu trúc gọi là 1 trường, có dạng khai báo như các biến: **KDL tên\_trường**;

## 2. Truy cập đến các thành phần của cấu trúc

Cú pháp: **tên\_biến\_cấu\_trúc.tên\_trường**

Nếu thành phần là một cấu trúc: **tên\_biến\_cấu\_trúc.tên\_trường\_cấu\_trúc.tên\_trường**

Ví dụ:

- **an.MaSo, binh.HoTen, binh.SoDT, an.DiaChi.**
- **an.NgaySinh.Thang, binh.NgaySinh.Nam.**

## 3. Nhập, xuất dữ liệu cho biến cấu trúc

### a. Nhập giá trị cho từng thành phần của cấu trúc

- **cin >> binh.MaSo;** Nhập mã số cho nhân viên binh
- **cin >> an.NgaySinh.Nam** Nhập năm sinh cho nhân viên an
- **gets\_s(an.DiaChi)** Nhập địa chỉ của nhân viên an

### b. Xuất giá trị của từng thành phần cấu trúc

- **cout << binh.MaSo;** Xuất mã số của nhân viên binh ra màn hình
- **cout << an.NgaySinh.Nam** Xuất năm sinh của nhân viên an
- **cout << an.DiaChi** Xuất địa chỉ của nhân viên an

## 4. Khởi đầu (gán) giá trị biến cấu trúc

Khởi đầu cho từng thành phần của cấu trúc, theo thứ tự.

- **NgayThang ns = { 31, 12, 1988 };**
- **NhanVien an = { 10, "Nguyen Van An", "1C NCTru", { 5, 12, 1990 }, 1234567890 };**
- **NhanVien binh = { 25, "Pham Binh", "100B BTXuan", { 24, 6, 1981 }, 1278934560 };**

## 5. Mảng cấu trúc

Mảng cấu trúc là mảng một chiều chứa các phần tử có cùng kiểu cấu trúc.

Khai báo: **TênKiểuCấuTrúc tên\_biến\_mảng[Kích\_thước];**

Ví dụ: NhanVien dsnv[1000];  
Định nghĩa kiểu mảng cấu trúc: **typedef TênKiểuCấuTrúc tên\_kiểu\_mảng[Kích\_thước];**  
Ví dụ: **typedef** NhanVien MangNV[1000];  
MangNV dsnv;

## 6. Truyền tham số

| Tham số hình thức | Tham số thực (đối số)                        |
|-------------------|--|
| Biến cấu trúc     | Giá trị cấu trúc của cùng kiểu               |
| Tên mảng cấu trúc | Tên mảng cấu trúc cùng kiểu, cùng kích thước |

## D. Hướng dẫn thực hành

*Phần này xây dựng chương trình thực hiện các thao tác trên kiểu cấu trúc với các chức năng cơ bản. Tiếp tục tổ chức chương trình theo thư viện và có/không có hệ thống menu.*

### Bài 1: Quản lý nhân viên

Một công ty quản lý nhân viên theo các thông tin:

- Mã nhân viên : một chuỗi có đúng 7 ký tự
- Họ và Chữ lót : chuỗi không quá 20 ký tự
- Tên: chuỗi không quá 7 ký tự
- Ngày sinh : từ 1 đến 2 ký số
- Tháng sinh : từ 1 đến 2 ký số
- Năm sinh : 4 ký số
- Địa chỉ : chuỗi không quá 20 ký tự
- Lương : một số thực dương

Viết chương trình tùy chọn thực hiện trên danh sách các nhân viên, với các chức năng:

0. Thoát khỏi chương trình
1. Nhập danh sách nhân viên
2. Xem danh sách nhân viên
3. Thêm 1 nhân viên
4. Xóa một nhân viên
5. Sửa thông tin nhân viên
6. Tìm nhân viên theo mã số
7. Tìm nhân viên theo tên
8. Sắp danh sách sinh viên tăng dần theo mã nhân viên
9. Sắp danh sách nhân viên tăng dần theo tên, theo họ và lương
10. Tính tổng lương tháng
11. Liệt kê các nhân viên có lương tháng  $\geq x$  (x cho trước)
12. Liệt kê các nhân viên có năm sinh trong khoảng [u,v]

Bước 1. Tạo 1 dự án Win32 Console Application mới. Đặt tên là **Lab08\_D\_Bai1**

Bước 2. Tạo cấu trúc cho chương trình như bài 1 mục D lab 7 (phần lõi tối thiểu chạy được của chương trình)

Tức là ta có kết quả chương trình ở bước này như sau :

- Tập tin **thuvien.h** : Rỗng
- Tập tin **menu.h** : Rỗng
- Tập tin **program.cpp** có nội dung như sau :

```
#include <iostream>
```

```

#include <conio.h>
using namespace std;

#include "thuvien.h"
#include "menu.h"

void ChayChuongTrinh();

int main()
{
    ChayChuongTrinh();
    return 1;
}
void ChayChuongTrinh()
{
    _getch();
}

```

Nhấn Ctrl + F5 để chạy chương trình, sửa lỗi nếu có.

### **Bước 3:**

Bước này ta định nghĩa kiểu dữ liệu mới; tổ chức và vận hành hệ thống menu.

#### **- Trong tập tin *thuvien.h* :**

Ta bổ sung định nghĩa hằng cho kích thước khai báo mảng 1 chiều, các kiểu dữ liệu cấu trúc : NhanVien, NgayThangNam .

*//Định nghĩa hằng*

*//Định nghĩa hằng*

*#define MAX 100//kích thước khai báo mảng cấu trúc*

*//Định nghĩa kiểu dữ liệu mới : các kiểu cấu trúc*

*typedef unsigned short int Byte;*

*//Kiểu ngày tháng năm sinh*

*struct NgayThangNam*

*{*

*Byte ngaySinh;*

*Byte thangSinh;*

*unsigned int namSinh;*

*};*

*//Kiểu nhân viên*

*struct NhanVien*

*{*

*char maNV[8];*

*char hoLot[21];*

*char ten[8];*

*NgayThangNam ntns;*

*char diaChi[21];*

*double luong;*

*};*

*//Khai báo nguyên mẫu các hàm xử lý, nhập xuất*

*//bổ sung sau*

*//Định nghĩa các hàm xử lý, nhập xuất*

//bổ sung sau

- Trong tập tin *menu.h* :

// Khai báo nguyên mẫu các hàm xử lý menu

//bổ sung sau

// Định nghĩa các hàm xử lý menu

3.1 Định nghĩa hàm xuất danh sách chức năng ra màn hình

void XuatMenu()

```
{
    cout << "\n=====He thong chuc nang===== ";
    cout << "\n0. Thoat khoi chuong trinh";
    cout << "\n1. Nhap danh sach nhan vien";
    cout << "\n2. xem danh sach nhan vien";
    cout << "\n3. Them mot nhan vien vao cuoi danh sach";
    cout << "\n4. Xoa mot nhan vien theo ma nhan vien";
    cout << "\n5. Sua thong tin nhan vien theo ma nhan vien";
    cout << "\n6. Tim nhan vien theo ma so";
    cout << "\n7. Tim nhan vien theo ten";
    cout << "\n8. Sap danh sach nhan vien tang dan theo ma nhan vien";
    cout << "\n9. Sap danh sach nhan vien tang dan theo ten - ho - luong";
    cout << "\n10. Tinh tong luong thang";
    cout << "\n11. Liet ke cac nhan vien co luong >= x (nhap tu ban phim)";
    cout << "\n12. Liet ke cac nhan vien co nam sinh trong khoang [u,v]";
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.2 Định nghĩa hàm chọn một menu trong danh sách

// Input : soMenu = Số lượng menu có thể chọn.

// Output: Số thứ tự menu do người dùng nhập vào.

int ChonMenu(int soMenu)

```
{
    int stt;
    for (;;)
    {
        system("CLS");
        XuatMenu();
        cout << "\nNhap 1 so khong khoang [0,...," << soMenu << "] de chon chuc nang, stt = ";
        cin >> stt;
        if (0 <= stt && stt <= soMenu)
            break;
    }
    return stt;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.3 Định nghĩa hàm xử lý menu :

Các thao tác thực hiện trên cùng một đầu vào là mảng 1 chiều kiểu NhanVien, kích thước mảng n, nên ta bổ sung thêm biến mảng 1 chiều NhanVien a, số nguyên dương n làm đối của hàm **XuLyMenu**, ngoài tham số đã có là tham số menu.

void XuLyMenu(int menu, NhanVien a[MAX], int &n)

```
{
```

```

//khai bao bien
switch (menu)
{
case 0:
    system("CLS");
    cout << "\n0. Thoat khoi chuong trinh\n";
    break;
case 1:
    system("CLS");
    cout << "\n1. Nhap danh sach nhan vien";

    break;
case 2:
    system("CLS");
    cout << "\n2. Xem danh sach nhan vien";

    break;
case 3:
    system("CLS");
    cout << "\n3. Them mot nhan vien";
    break;
case 4:
    system("CLS");
    cout << "\n4. Xoa mot nhan vien";
    break;
case 5:
    system("CLS");
    cout << "\n5. Sua thong tin nhan vien";
    break;
case 6:
    system("CLS");
    cout << "\n6. Tim nhan vien theo ma so";
    break;
case 7:
    system("CLS");
    cout << "\n7. Tim nhan vien theo ten";
    break;
case 8:
    system("CLS");
    cout << "\n8. Sap danh sach nhan vien tang dan theo ma nhan vien";
    break;

case 9:
    system("CLS");
    cout << "\n9. Sap danh sach nhan vien tang dan theo ten - ho - luong";
    break;
case 10:
    system("CLS");
    cout << "\n10. Tinh tong luong thang";
    break;
case 11:

```

```

        system("CLS");
        cout << "\n11. Liet ke cac nhan vien co luong >= x ";
        break;
    case 12:
        system("CLS");
        cout << "\n12. Liet ke cac nhan vien co nam sinh trong khoang [u,v]";
        break;
    }
    _getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

### 3.4 Bổ sung khai báo nguyên mẫu các hàm tổ chức menu trong phần khai báo nguyên mẫu hàm

```

void XuatMenu();
int ChonMenu(int soMenu);
void XuLyMenu(int menu, NhanVien a[MAX], int &n);

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có

- Trong tập tin *program.cpp* :
- + Hàm *ChayChuongTrinh* ta cập nhật lại như sau :

```

void ChayChuongTrinh()
{
    int menu,
        soMenu = 12,
        n = 0;
    NhanVien a[MAX];
    do
    {
        menu = ChonMenu(soMenu);
        XuLyMenu(menu, a, n);
    } while (menu > 0);
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
 Kiểm tra sự vận hành của hệ thống menu.  
 Kiểm tra chức năng thoát khỏi chương trình (chọn 0).

### Bước 4 : Bổ sung vào chương trình các thao tác nhập xuất dữ liệu

Trong bước 4 này, ta làm công việc sau :

- Trong tập tin *program.cpp* , bổ sung các thư viện cần thiết
- Trong tập tin *thuvien.h* , soạn thảo các hàm nhập, xuất dữ liệu
- Trong tập tin *menu.h* bổ sung xử lý chức năng nhập dữ liệu, xem chuỗi trong hàm *XuLyMenu*.

- Trong tập tin *program.cpp* :  
 Bổ sung thư viện <string.h>, <iomanip>

- Trong tập tin *thuvien.h* :  
 Bổ sung các hàm nhập xuất dữ liệu

#### 4.1 Nhập dữ liệu :

```

//Ham Nhap 1 nhan vien tu ban phim
void Nhap_INV(NhanVien &p)

```

```

{
    cout << "\nMa Nhan Vien (dung 7 ky so) : ";
    _flushall();
    gets_s(p.maNV, 8);

    cout << "\nHo va chu lot: ";
    _flushall();
    gets_s(p.hoLot, 21);

    cout << "\nTen nhan vien: ";
    _flushall();
    gets_s(p.ten, 7);

    cout << "\nNgay sinh: ";
    _flushall();
    cin >> p.ntns.ngaySinh;

    cout << "\nThang sinh: ";
    _flushall();
    cin >> p.ntns.thangSinh;

    cout << "\nNam sinh: ";
    _flushall();
    cin >> p.ntns.namSinh;

    cout << "\nNhap dia chi: ";
    _flushall();
    gets_s(p.diaChi, 20);

    cout << "\nNhap luong: ";
    _flushall();
    cin >> p.luong;
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

//Hàm nhập danh sách nhân viên

```

void Nhap_DS NV(NhanVien a[MAX], int &n)
{
    int i;
    cout << "\nNhap n = ";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        system("CLS");
        cout << "\nNhap thông tin cho nhân viên thứ " << i + 1 << " : ";
        Nhap_1NV(a[i]);
    }
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

#### 4.2 Hàm xuất dữ liệu ra màn hình

//Xuất ke ngang



```

void XuatKeNgang()
{
    int i;
    cout << "\n";
    cout << setiosflags(ios::left)
        << ':';
    for (i = 1; i <= 76; i++)
        cout << NGANGDOI;
    cout << ':';
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```

//Xuat tieu de
void XuatTieuDe()
{
    XuatKeNgang();
    cout << endl;
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << "Ma NV"
        << ':'
        << setw(16) << "Ho"
        << setw(8) << "Ten"
        << ':'
        << setw(12) << "NTN sinh"
        << ':'
        << setw(16) << "Dia Chi"
        << ':'
        << setw(12) << "Luong"
        << ':';
    XuatKeNgang();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```

//Xuat 1 nhan vien
void Xuat_1NV(NhanVien p)
{
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << p.maNV
        << ':'
        << setw(16) << p.hoLot
        << setw(8) << p.ten
        << ':'
        << setw(2) << p.ntns.ngaySinh
        << '/'
        << setw(2) << p.ntns.thangSinh
        << '/'
        << setw(6) << p.ntns.namSinh
        << ':'
        << setw(16) << p.diaChi
        << ':'
        << setw(12) << setiosflags(ios::fixed) << setprecision(2) << p.luong
}

```

```

        << "!\n";
    }

    Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```

```

//Xuat DS nhan vien
void Xuat_DSNV(NhanVien a[MAX], int n)
{
    int i;
    XuatTieuDe();
    for (i = 0; i < n; i++)
    {
        cout << endl;
        Xuat_1NV(a[i]);
    }
    XuatKeNgang();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

#### 4.3 Bổ sung nguyên mẫu các hàm :

```

void Nhap_1NV(NhanVien &p);
void Nhap_1NV(NhanVien &p);
void XuatKeNgang();
void XuatTieuDe();
void Xuat_1NV(NhanVien p);
void Xuat_DSNV(NhanVien a[MAX], int n);

```

#### - Trong tập tin *menu.h* :

Bổ sung xử lý chức năng nhập dữ liệu vào case 1, xem dữ liệu vào case 2 (Các case từ 3 đến 12 giữ nguyên)

```

void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    switch (menu)
    {
        case 0:
            system("CLS");
            cout << "\n0. Thoat khoi chuong trinh\n";
            break;

        case 1:
            system("CLS");
            cout << "\n1. Nhap danh sach nhan vien";
            Nhap_DSNV(a, n);
            system("CLS");
            Xuat_DSNV(a, n);
            cout << "\nSo nhan vien trong danh sach : n = " << n;
            break;

        case 2:
            system("CLS");
            cout << "\n2. Xem danh sach nhan vien";
            Xuat_DSNV(a, n);

```

```

        cout << "\nSo nhan vien trong danh sach : n = " << n;
        break;
    //...

}

_getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra kết quả thực hiện các chức năng 1, 2.

#### Bước 5 :

Ta có nhận xét là việc nhập dữ liệu cho danh sách nhân viên (mảng cấu trúc) từ bàn phím rất mất thời gian, đặc biệt là phải lặp lại cho mỗi lần chạy chương trình, nên ta cần có một cách phù hợp để tiết kiệm thời gian. Một trong các cách có thể dùng là khởi tạo dữ liệu ban đầu cho danh sách, sau đó muốn tạo lại danh sách ta có thể chọn chức năng 1 (nhập danh sách nhân viên từ bàn phím)

Ta sửa lại như sau :

- Trong tập tin menu.h, để cho trực quan hàm XuatMenu ta cập nhật lại như sau :

```

void XuatMenu()
{
    cout << "\n=====He thong chuc nang===== ";
    cout << "\n0. Thoat khỏi chương trình";
    cout << "\n1. Nhập danh sách nhân viên";
    cout << "\n2. xem danh sách nhân viên";
    cout << "\n3. Thêm một nhân viên vào cuối danh sách";
    cout << "\n4. Xóa một nhân viên theo mã nhân viên";
    cout << "\n5. Sửa thông tin nhân viên theo mã nhân viên";
    cout << "\n6. Tìm nhân viên theo mã số";
    cout << "\n7. Tìm nhân viên theo tên";
    cout << "\n8. Sắp danh sách nhân viên tăng dần theo mã nhân viên";
    cout << "\n9. Sắp danh sách nhân viên tăng dần theo tên - họ - lương";
    cout << "\n10. Tính tổng lương tháng";
    cout << "\n11. Liệt kê các nhân viên có lương >= x ";
    cout << "\n12. Liệt kê các nhân viên có năm sinh trong khoảng [u,v]";
    cout << "\n===== ";
    cout << "\n  (Dữ liệu đã được khởi tạo, chọn 1 nếu muốn tạo lại dữ liệu.)";
    cout << "\n===== ";
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

- Trong tập tin program.cpp, ta khởi tạo dữ liệu cho mảng a và kích thước n trong hàm ChayChuongTrinh:

```

void ChayChuongTrinh()
{
    int menu,
        soMenu = 12,
        n = 10; //khởi đầu n

    NhanVien a[MAX] = //khởi đầu a
    {
        { "1234507", "Nguyen Van", "Tan", { 1, 1, 1990 }, "Da Lat", 20345678.0 },
        { "1205507", "Tran Minh", "Hoang", { 10, 10, 1980 }, "Qui Nhon", 14345678.0 },

```

```

        { "2234067", "Truong", "Hoang", { 10,12, 1980 }, "Da Lat", 10342678.0 },
        { "2134167", "Le Tuan", "Ban", { 10, 1, 1975 }, "Quang Nam", 18349678.0 },
        { "1135067", "Tran Minh", "Ban", { 5, 6, 1977 }, "Phu Yen", 12345978.0 },
        { "1034007", "Truong", "Hoang", { 21, 12, 1987 }, "Da Lat", 15348678.0 },
        { "1130160", "Vo Tuan", "Trong", { 1, 11, 1985 }, "Qui Nhon", 20341178.0 },
        { "1235567", "Hoang Van", "Tan", { 15, 12, 1970 }, "Phu Yen", 22300970.0 },
        { "1034327", "Tran Van", "Vu", { 11, 11, 1982 }, "Da Lat", 16348888.0 },
        { "1034467", "Hoang Trong", "Ban", { 12, 1, 1980 }, "Nha Trang", 14365698.0 }
    };

do
{
    menu = ChonMenu(soMenu);
    XuLyMenu(menu, a, n);
} while (menu > 0);
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra chức năng 2 với bộ dữ liệu khởi tạo

Trong các bước tiếp theo:

- Ta sử dụng bộ dữ liệu khởi tạo. Khi nào cần thay đổi bộ dữ liệu khác, ta chọn chức năng 1 để nhập lại dữ liệu từ bàn phím
- Bổ sung lần lượt từng chức năng (từ 3 đến 12) vào chương trình :
  - o Lần lượt soạn thảo từng hàm chức năng trong tập tin *thuvien.h* .
  - o Bổ sung xử lý chức năng tương ứng trong hàm *XuLyMenu* của *menu.h*,

Bước 6: Bổ sung chức năng 3 (thêm một nhân viên vào cuối danh sách) vào chương trình

- Trong *thuvien.h* viết hàm thêm nhân viên

6.1 Hàm thêm nhân viên vào cuối danh sách:

*//Thêm nhân viên vào cuối danh sách*

```

void ThemNhanVien_Cuoi(NhanVien a[MAX], int &n, NhanVien p)
{
    a[n++] = p;
}

```

6.2 Khai báo nguyên mẫu :

```

void ThemNhanVien_Cuoi(NhanVien a[MAX], int &n, NhanVien p);

```

- Trong menu.h, bổ sung xử lý hàm thêm nhân viên vào case 3 của hàm XuLyMenu :

```

void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    switch (menu)
    {
        //...
        case 3:
            system("CLS");
            cout << "\n3. Them mot nhan vien";
            cout << "\nNhap thong tin nhan vien can chen vao cuoi danh sach:\n";
            Nhap_1NV(p);
            system("CLS");
            cout << "\nDanh sach ban dau :\n";
            Xuat_DS NV(a, n);
            cout << "\nSo nhan vien trong danh sach : n = " << n;

```

```

        ThemNhanVien_Cuoi(a, n, p);
        cout << "\nDanh sach ket qua :\n";
        Xuat_DS NV(a, n);
        cout << "\nSo nhan vien trong danh sach sau khi them : n = " << n;
        break;
    //...
}

_getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra chức năng 3.

Bước 7: Bổ sung chức năng 4 (Xóa một nhân viên theo mã nhân viên) vào chương trình

- Trong **thuvien.h** viết hàm xóa nhân viên

7.1 Hàm xóa nhân viên (theo mã nhân viên) :

//Xóa nhân viên theo mã số

```

void XoaNhanVien_MaNV(NhanVien a[MAX], int &n, char maNV[8])
{
    int i,j;
    for (i = 0; i < n; i++)
        if (strcmp(a[i].maNV, maNV) == 0)
            break;
    if (i == n)
    {
        cout << "\nKhong co nhan vien nao trong danh sach co ma so " << maNV;
        return;
    }
    for (j = i + 1; j < n; j++)
        a[j - 1] = a[j];
    n--;
}

```

7.2 Khai báo nguyên mẫu :

```
void XoaNhanVien_MaNV(NhanVien a[MAX], int &n, char maNV[8]);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm xóa nhân viên vào case 4 :

```

void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8];
    switch (menu)
    {
        //...
        case 4:
            system("CLS");
            cout << "\n4. Xoa mot nhan vien theo ma so";
            cout << "\nNhap ma nhan vien can xoa :";
            cin >> maNV;

```

```

        cout << "\nDanh sach hien hanh :\n";
        Xuat_DS NV(a, n);
        cout << "\nSo nhan vien trong danh sach : n = " << n;
        XoaNhanVien_MaNV(a, n, maNV);

        cout << "\nDanh sach ket qua :\n";
        Xuat_DS NV(a, n);
        cout << "\nSo nhan vien trong danh sach sau khi xoa : n = " << n;
        break;
    //...
}

_getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra chức năng 4.

Bước 8: Bổ sung chức năng 5 (Sua thông tin nhân viên theo mã nhân viên) vào chương trình :

- Trong **thuvien.h** viết hàm sửa thông tin nhân viên

8.1 Hàm sửa thông tin nhân viên (theo mã nhân viên) :

//Sua thông tin nhan vien theo ma so

```

void SuaNhanVien_MaNV(NhanVien a[MAX], int &n, char maNV[8])
{
    int i;
    char tl;
    for (i = 0; i < n; i++)
        if (strcmp(a[i].maNV, maNV) == 0)
            break;
    if (i == n)
    {
        cout << "\nKhong co nhan vien nao trong danh sach co ma so " << maNV;
        return;
    }

    cout << "\nSua thông tin nhan vien co ma so : " << maNV;
    cout << "\nCo sua ma nhan vien khong ? nhan k neu khong: ";
    tl = _getch();
    if (tl != 'k' && tl != 'K')
    {
        cout << "\nMa Nhan Vien : ";
        _flushall();
        gets_s(a[i].maNV, 8);
    }

    cout << "\nCo sua ho va chu lot khong ? nhan k neu khong: ";
    tl = _getch();
    if (tl != 'k' && tl != 'K')
    {
        cout << "\nHo va chu lot: ";
        _flushall();
        gets_s(a[i].hoLot, 21);
    }
}

```

```

cout << "\nCo sua ten khong ? nhan k neu khong: ";
tl = _getch();
if (tl != 'k' && tl != 'K')
{
    cout << "\nTen nhan vien: ";
    _flushall();
    gets_s(a[i].ten, 8);
}

cout << "\nCo sua ngay sinh khong ? nhan k neu khong: ";
tl = _getch();
if (tl != 'k' && tl != 'K')
{
    cout << "\nNgay sinh: ";
    _flushall();
    cin >> a[i].ntns.ngaySinh;
}

cout << "\nCo sua thang sinh khong ? nhan k neu khong: ";
tl = _getch();
if (tl != 'k' && tl != 'K')
{
    cout << "\nThang sinh: ";
    _flushall();
    cin >> a[i].ntns.thangSinh;
}

cout << "\nCo sua nam sinh khong ? nhan k neu khong: ";
tl = _getch();
if (tl != 'k' && tl != 'K')
{
    cout << "\nNam sinh: ";
    _flushall();
    cin >> a[i].ntns.namSinh;
}

cout << "\nCo sua dia chi khong ? nhan k neu khong: ";
tl = _getch();
if (tl != 'k' && tl != 'K')
{
    cout << "\nNhap dia chi: ";
    _flushall();
    gets_s(a[i].diaChi, 20);
}

cout << "\nCo sua luong khong ? nhan k neu khong: ";
tl = _getch();
if (tl != 'k' && tl != 'K')
{
    cout << "\nNhap luong: ";
    _flushall();
    cin >> a[i].luong;
}
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

## 8.2 Khai báo nguyên mẫu :

```
void SuaNhanVien_MaNV(NhanVien a[MAX], int &n, char maNV[8]);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm sửa thông tin nhân viên vào case 5 :

```
void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8];
    switch (menu)
    {
        //...
        case 5:
            system("CLS");
            cout << "\n5. Sua thong tin nhan vien";
            cout << "\nDanh sach hien hanh :\n";
            Xuat_DSNV(a, n);

            cout << "\nNhap ma nhan vien can sua :";
            cin >> maNV;
            SuaNhanVien_MaNV(a, n, maNV);

            cout << "\nDanh sach ket qua :\n";
            Xuat_DSNV(a, n);
            break;
        //...
    }
    _getch();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra chức năng 5.

Bước 9: Bổ sung chức năng 6 (tìm nhân viên theo mã nhân viên) vào chương trình :

- Trong **thuvien.h**

### 9.1 Hàm tìm nhân viên (theo mã nhân viên) :

```
//Tim nhan vien theo ma so
//Khong co: tra ve -1; co : tra ve i (chi so bieu dien nhan vien tim duoc trong danh sach)
int TimNhanVien_MaNV(NhanVien a[MAX], int n, char maNV[8])
{
    int i, kq = -1;
    for (i = 0; i < n; i++)
        if (strcmp(a[i].maNV, maNV) == 0)
        {
            kq = i;
            break;
        }
    return kq;
}
```



Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

## 9.2 Khai báo nguyên mẫu :

```
int TimNhanVien_MaNv(NhanVien a[MAX], int n, char maNV[8]);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung khai báo biến kq kiểu int để lưu trữ kết quả tìm kiếm

+ Bổ sung xử lý hàm tìm nhân viên vào case 6 :

```
void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8];
    int kq;
    switch (menu)
    {
        //...
        case 6:
            system("CLS");
            out << "\n6. Tim nhan vien theo ma so";
            cout << "\nDanh sach hien hanh :\n";
            Xuat_DSNV(a, n);
            cout << "\nNhap ma nhan vien can tim :";
            cin >> maNV;
            kq = TimNhanVien_MaNv(a, n, maNV);

            if (kq == -1)
                cout << "\nKhong co NV nao trong danh sach co ma so : " << maNV;
            else
            {
                cout << "\nthong tin nhan vien co ma so : " << maNV << " :\n";
                XuatTieuDe();
                Xuat_1NV(a[kq]);
            }
            break;
        //...
    }
    _getch();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra chức năng 6.

Bước 10: Bổ sung chức năng 7 (tìm nhân viên theo tên) vào chương trình :

- Trong **thuvien.h**

10.1 Hàm tìm nhân viên (theo tên nhân viên) :

//Tìm nhân viên theo tên

```
void TimNhanVien_Ten(NhanVien a[MAX], int n, char ten[7])
```

```
{
    int i,
        dem = 0; //luu tru co bao nhieu nhan vien co ten nhu vay
    for (i = 0; i < n; i++)
        if (_stricmp(a[i].ten, ten) == 0)
            dem++;
}
```

```

if (!dem)
    cout << "\nTrong danh sach khong co nhan vien nao co ten la " << ten;
else
{
    cout << "\nCo " << dem << " nhan vien ten " << ten
        << ", voi thong tin chi tiet nhu sau :\n";
    XuatTieuDe();
    for (i = 0; i < n; i++)
        if (_strcmpr(a[i].ten, ten) == 0)
        {
            cout << endl;
            Xuat_1NV(a[i]);
        }
    XuatKeNgang();
}
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

10.2 Khai báo nguyên mẫu :

```
void TimNhanVien_Ten(NhanVien a[MAX], int n, char ten[7]);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung khai báo chuỗi lưu trữ tên nhân viên cần tìm

+ Bổ sung xử lý hàm tìm nhân viên theo tên vào case 7:

```

void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8], ten[7];
    int kq;
    switch (menu)
    {
        //...
        case 7:
            system("CLS");
            cout << "\n7. Tim nhan vien theo ten";
            cout << "\nDanh sach hien hanh :\n";
            Xuat_DS NV(a, n);
            cout << "\nNhap ten nhan vien can tim :";
            cin >> ten;
            TimNhanVien_Ten(a, n, ten);
            break;
        //...
    }

    _getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra chức năng 7.

Bước 11: Bổ sung chức năng 8 (sắp danh sách tăng dần theo mã nhân viên) vào chương trình :

- Trong *thuvien.h*

11.1 Hàm sắp danh sách tăng dần theo mã nhân viên:

//Sap tang theo Ma Nhan Vien

```
void SapTang_MaNV(NhanVien a[MAX], int n)
{
    int i, j;
    //Tang theo ma nhan vien
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (_stricmp(a[i].maNV, a[j].maNV) > 0)
                HoanVi(a[i], a[j]);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

11.2 Hoàn hoán vị :

```
void HoanVi(NhanVien &p, NhanVien &q)
{
    NhanVien t;
    t = p;
    p = q;
    q = t;
}
```

11.3 Khai báo nguyên mẫu :

```
void SapTang_MaNV(NhanVien a[MAX], int n);
```

```
void HoanVi(NhanVien &p, NhanVien &q);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm sắp tăng theo tên vào case 8:

```
void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8], ten[7];
    int kq;
    switch (menu)
    {
        //...
        case 8:
            system("CLS");
            cout << "\n8. Sap danh sach nhan vien tang dan theo ma nhan vien";
            cout << "\nDanh sach ban dau :\n";
            Xuat_DS NV(a, n);
            SapTang_MaNV(a, n);
            cout << "\nDanh sach ket qua :\n";
            Xuat_DS NV(a, n);
            break;
        //...
    }

    _getch();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra chức năng 8.

Bước 12: Bổ sung chức năng 9 (sắp danh sách tăng dần theo tên-họ-lương) vào chương trình :

- Trong **thuvien.h**

12.1 Hàm sắp danh sách tăng dần theo tên-họ-lương:

//Sap tang theo Ten-Ho-Luong

```
void SapTang_Ten_Ho_Luong(NhanVien a[MAX], int n)
{
    int i, j;
    //Tang theo ten
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (_stricmp(a[i].ten, a[j].ten) > 0)
                HoanVi(a[i], a[j]);

    //Ten trung, tang theo ho
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (_stricmp(a[i].ten, a[j].ten) == 0)
                if (_stricmp(a[i].hoLot, a[j].hoLot) > 0)
                    HoanVi(a[i], a[j]);

    //Ten va ho trung, tang theo luong
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (_stricmp(a[i].ten, a[j].ten) == 0 && _stricmp(a[i].hoLot, a[j].hoLot) == 0)
                if (a[i].luong > a[j].luong)
                    HoanVi(a[i], a[j]);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

12.2 Khai báo nguyên mẫu :

```
void SapTang_Ten_Ho_Luong(NhanVien a[MAX], int n);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm sắp tăng tên-họ-lương theo tên vào case 9:

```
void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8], ten[7];
    int kq;
    switch (menu)
    {
        //...
        case 9:
            system("CLS");
            cout << "\n9. Sap danh sach nhan vien tang dan theo ten-ho luong";
            cout << "\nDanh sach ban dau :\n";
            Xuat_DS NV(a, n);
            SapTang_Ten_Ho_Luong(a, n);
            cout << "\nDanh sach ket qua :\n";
            Xuat_DS NV(a, n);
    }
}
```

```

        break;
    //...
}

_getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra chức năng 9.

Bước 13: Bổ sung chức năng 10 (tính tổng lương) vào chương trình :

- Trong **thuvien.h**

13.1 Hàm tính tổng lương:

```

double TinhTongLuong(NhanVien a[MAX], int n)
{
    int i;
    double tong = 0;
    for (i = 0; i < n; i++)
        tong += a[i].luong;
    return tong;
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

13.2 Khai báo nguyên mẫu :

```
double TinhTongLuong(NhanVien a[MAX], int n);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm tính tổng lương theo tên vào case 10:

```

void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8], ten[7];
    int kq;
    switch (menu)
    {
        //...
        case 10:
            system("CLS");
            cout << "\n10. Tinh tong luong thang";
            cout << "\nDanh sach hien hanh :\n";
            Xuat_DSNV(a, n);
            cout << "\nTong luong thang: tong = " << TinhTongLuong(a, n);
            break;
        //...
    }

    _getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra chức năng 10.

Bước 14: Bổ sung chức năng 11 (liệt kê các nhân viên có lương  $\geq x$ ) vào chương trình :

- Trong **thuvien.h**

14.1 Hàm liệt kê các nhân viên có lương  $\geq x$ :

```
void Xuat_DSNV_Luong_KhongNhoHon_x(NhanVien a[MAX], int n, double x)
{
    int i, kq = 0;
    for (i = 0; i < n; i++)
        if (a[i].luong >= x)
            kq++;
    if (!kq)
        cout << "\nKhong co nhan vien nao co luong >= " << x;
    else
    {
        cout << "\n\nCo " << kq << " nhan vien co luong >= " << x
            << ", voi thong tin chi tiet nhu sau:";
        XuatTieuDe();
        for (i = 0; i < n; i++)
        {
            if (a[i].luong >= x)
            {
                cout << endl;
                Xuat_1NV(a[i]);
            }
        }
        XuatKeNgang();
    }
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

14.2 Khai báo nguyên mẫu :

```
void Xuat_DSNV_Luong_KhongNhoHon_x(NhanVien a[MAX], int n, double x);
```

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm liệt kê các nhân viên có lương  $\geq x$  vào case 11:

```
void XuLyMenu(int menu, NhanVien a[MAX], int &n)
{
    //khai bao bien
    char maNV[8], ten[7];
    int kq;
    switch (menu)
    {
        //...
        case 11:
            system("CLS");
            cout << "\n11. Liet ke cac nhan vien co luong >= x (nhap tu ban phim)";
            cout << "\nNhap x = ";
            cin >> x;
            cout << "\nDanh sach hien hanh :\n";
            Xuat_DSNV(a, n);
            Xuat_DSNV_Luong_KhongNhoHon_x(a, n, x);
            break;
        //...
    }
}
```

```
_getch();
```

```
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra chức năng 11.

Bước 15: Bổ sung chức năng 12 (liệt kê các nhân viên có năm sinh trong khoảng [u,v],  $u \leq v$ ) vào chương trình :

- Trong **thuvien.h**

15.1 Hàm liệt kê các nhân viên có năm sinh trong khoảng [u,v],  $u \leq v$

**void** Xuat\_DSNV\_NamSinh\_u\_v(**NhanVien** a[**MAX**], **int** n, **unsigned int** u, **unsigned int** v)

```
{
    int i, kq = 0;
    for (i = 0; i < n; i++)
        if (u <= a[i].ntns.namSinh && a[i].ntns.namSinh <= v)
            kq++;
    if (!kq)
        cout << "\nKhong co nhan vien nao co nam sinh trong khoang [" << u << ", " << v << "]" ;
    else
    {
        cout << "\n\nCo " << kq << " nhan vien co nam sinh trong khoang [" << u
        << ", ..., " << v << "] : ";
        XuatTieuDe();
        for (i = 0; i < n; i++)
        {
            if (u <= a[i].ntns.namSinh && a[i].ntns.namSinh <= v)
            {
                cout << endl;
                Xuat_1NV(a[i]);
            }
        }
        XuatKeNgang();
    }
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

14.2 Khai báo nguyên mẫu :

**void** Xuat\_DSNV\_NamSinh\_u\_v(**NhanVien** a[**MAX**], **int** n, **unsigned int** u, **unsigned int** v);

- Trong hàm XuLyMenu của tập tin menu.h :

+ Bổ sung xử lý hàm liệt kê các nhân viên có năm sinh trong khoảng [u,v],  $u \leq v$  vào case 12:

**void** XuLyMenu(**int** menu, **NhanVien** a[**MAX**], **int** &n)

```
{
    //khai bao bien
    char maNV[8], ten[7];
    int kq;
    switch (menu)
    {
        //...
        case 12:
            system("CLS");
            cout << "\n12. Liet ke cac nhan vien co nam sinh trong khoang [u,v], u<=v";
            cout << "\nNhap u = ";
            cin >> u;
```

```

        cout << "\nNhap v = ";
        cin >> v;
        system("CLS");

        cout << "\nDanh sach hien hanh :\n";
        Xuat_DSNV(a, n);
        Xuat_DSNV_NamSinh_u_v(a, n, u, v);
        break;
    //...
}

_getch();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra chức năng 12.

Kiểm tra các chức năng chương trình – Kết thúc chương trình.

### Ghi chú:

Việc nhập dữ liệu cho mảng cấu trúc từ bàn phím nhiều lần rất mất thời gian, nên để thực hiện chức năng 1, ta có thể thay thế các hàm nhập dữ liệu cho danh sách nhân viên từ bàn phím như đã dùng bằng các hàm tạo dữ liệu sau đây:

//Chen 1 nhan vien

```

void Chen_NV(char maNV[8], char hoLot[15], char ten[7], Byte ngaySinh, Byte thangSinh,
unsigned int namSinh, char diaChi[15], double luong, NhanVien a[MAX], int &n)

```

```

{
    if (n < MAX)
    {
        strcpy_s(a[n].maNV, 8, maNV);
        strcpy_s(a[n].hoLot, 15, hoLot);
        strcpy_s(a[n].ten, 7, ten);
        a[n].ntns.ngaySinh = ngaySinh;
        a[n].ntns.thangSinh = thangSinh;
        a[n].ntns.namSinh = namSinh;
        strcpy_s(a[n].diaChi, 15, diaChi);
        a[n].luong = luong;
        n++;
    }
}

```

//Tao Danh Sach nhan vien

```

void TaoDanhSachNhanVien(NhanVien a[MAX], int &n)

```

```

{
    Chen_NV("1234507", "Nguyen Van", "Tan", 1, 1, 1990, "Da Lat", 20345678.0, a, n);
    Chen_NV("1205507", "Tran Minh", "Hoang", 10, 10, 1980, "Qui Nhon", 14345678.0, a, n);
    Chen_NV("2234067", "Truong", "Hoang", 10, 12, 1980, "Da Lat", 10342678.0, a, n);
    Chen_NV("2134167", "Le Tuan", "Ban", 10, 1, 1975, "Quang Nam", 18349678.0, a, n);
    Chen_NV("1135067", "Tran Minh", "Ban", 5, 6, 1977, "Phu Yen", 12345978.0, a, n);
    Chen_NV("1034007", "Truong", "Hoang", 21, 12, 1987, "Da Lat", 15348678.0, a, n);
    Chen_NV("1130160", "Vo Tuan", "Trong", 1, 11, 1985, "Qui Nhon", 20341178.0, a, n);
    Chen_NV("1235567", "Hoang Van", "Tan", 15, 12, 1970, "Phu Yen", 22300970.0, a, n);
    Chen_NV("1034327", "Tran Van", "Vu", 11, 11, 1982, "Da Lat", 16348888.0, a, n);
    Chen_NV("1034467", "Hoang Trong", "Ban", 12, 1, 1980, "Nha Trang", 14365698.0, a, n);
    Chen_NV("1134467", "Van Quoc", "Cuong", 1, 10, 1960, "Nha Trang", 24300000.0, a, n);
    Chen_NV("1533407", "Truong", "Hoang", 1, 12, 1967, "Da Nang", 16340000.0, a, n);
}

```



## Bài 2 : Số phân số

$$Q = \left\{ x = \frac{tu}{mau} : tu \in Z, mau \in N^* \right\}$$

Viết chương trình tùy chọn thực hiện trên các số phân số với các chức năng sau :

0. Thoát khỏi chương trình
1. Nhập phân số
2. Xem phân số
3. Rút gọn phân số
4. Cộng 2 phân số
5. Trừ phân số
6. Nhân 2 phân số
7. Chia phân số
8. So sánh 2 phân số

Tạo project với tên : **Lab08\_D\_Bai2\_PhanSo**.

Tổ chức chương trình tùy chọn menu như bài 1.

**Tham khảo tập tin [thuvien.h](#) sau đây :**

```
//Khai bao hang : khong co
//Định nghĩa Kiểu dữ liệu : PhanSo
struct PhanSo
{
    int tu;
    unsigned int mau;
};

//Khai bao nguyen mau cac ham chuc nang, nhap xuat
PhanSo Nhap_PS();
void XuatPS(PhanSo a);
unsigned int Tinh_GTTD(int x);
unsigned int UCLN(unsigned int a, unsigned int b);
unsigned int BCNN(unsigned int a, unsigned int b);
void RutGon(PhanSo &a);
void QuyDong_PS(PhanSo &a, PhanSo &b);
PhanSo CongPhanSo(PhanSo a, PhanSo b);
PhanSo TruPhanSo(PhanSo a, PhanSo b);
PhanSo NhanPhanSo(PhanSo a, PhanSo b);
PhanSo ChiaPhanSo(PhanSo a, PhanSo b);
int SoSanhPhanSo(PhanSo a, PhanSo b);

//Định nghĩa các hàm

//Hàm nhập phân số
PhanSo Nhap_PS()
{
    PhanSo a;
    cout << "\nNhập tử số : ";
    cin >> a.tu;
    do
    {
        cout << "\nNhập mẫu số (> 0) : ";
```

```

        cin >> a.mau;
    } while (a.mau <= 0);
    return a;
}
//Ham xuất phân số
void XuatPS(PhanSo a)
{
    if (a.tu == 0)
    {
        cout << 0;
        return;
    }
    if (a.mau == 1)
    {
        cout << a.tu;
        return;
    }

    cout << a.tu << '/' << a.mau;
}

//gia tri tuyet doi
unsigned int Tinh_GTTD(int x)
{
    return (x >= 0 ? x : -x);
}

//Tinh ước chung lớn nhất 2 số nguyên dương
unsigned int UCLN(unsigned int a, unsigned int b)
{
    unsigned int r;
    while (b > 0)
    {
        r = a % b;
        a = b;
        b = r;
    }
    return a;
}

//Bội chung nhỏ nhất 2 số nguyên dương
unsigned int BCNN(unsigned int a, unsigned int b)
{
    return (a*b) / UCLN(a, b);
}

//Rút gọn phân số
void RutGon(PhanSo &a)
{
    unsigned int d;
    d = UCLN(Tinh_GTTD(a.tu), a.mau);
    a.mau /= d;
    a.tu = a.tu / (int)d;
}

```

```

}

//quy dong mau so 2 phan so
void QuyDongMauSo_2PS(PhanSo &a, PhanSo &b)
{
    unsigned int d;
    RutGon(a);
    RutGon(b);
    d = BCNN(a.mau, b.mau);
    a.tu = a.tu * (d / a.mau);
    b.tu = b.tu * (d / b.mau);
    a.mau = b.mau = d;
}

//c = a+b
PhanSo CongPhanSo(PhanSo a, PhanSo b)
{
    PhanSo c;
    QuyDong_PS(a, b);
    c.tu = a.tu + b.tu;
    c.mau = a.mau;
    RutGon(c);
    return c;
}

//c = a - b
PhanSo TruPhanSo(PhanSo a, PhanSo b)
{
    PhanSo c;
    QuyDong_PS(a, b);
    c.tu = a.tu - b.tu;
    c.mau = a.mau;
    RutGon(c);
    return c;
}

//c = a * b
PhanSo NhanPhanSo(PhanSo a, PhanSo b)
{
    PhanSo c;
    RutGon(a);
    RutGon(b);
    c.tu = a.tu * b.tu;
    c.mau = a.mau * b.mau;
    RutGon(c);
    return c;
}

//c = a / b
PhanSo ChiaPhanSo(PhanSo a, PhanSo b)
{
    PhanSo c;
    RutGon(a);
    RutGon(b);
    int tu;

```

```

int mau;
tu = a.tu * b.mau;
mau = a.mau * b.tu;
if (mau < 0)
{
    tu = -tu;
    mau = -mau;
}
c.tu = tu;
c.mau = (unsigned int) mau;

RutGon(c);
return c;
}
//So sanh 2 phan so
//kq = - 1: a < b
//kq = 0 : a = b
//kq = 1 : a > b
int SoSanhPhanSo(PhanSo a, PhanSo b)
{
    int kq;
    QuyDong_PS(a, b);
    if (a.tu < b.tu)
        kq = -1;
    else
        if (a.tu == b.tu)
            kq = 0;
        else
            kq = 1;
    return kq;
}

```

Bài 3:

Bảng điểm môn “**Lập trình cấu trúc**” bao gồm các thông tin :

- Mã sinh viên: một chuỗi có đúng 7 ký tự (số, chữ)
- Họ và chữ lót sinh viên : chuỗi có không quá 15 ký tự
- Tên sinh viên : chuỗi không quá 6 ký tự
- Năm sinh : số nguyên dương (4 chữ số)
- Giới tính : không quá 3 ký tự
- Quê quán (tỉnh, thành phố) : chuỗi không quá 15 ký tự
- Điểm tổng kết : Điểm tổng kết: một số thực trong đoạn [0,...,10]

Điểm tổng kết môn học được tính từ các cột điểm :

- Điểm bài lab : chiếm 20% điểm tổng kết
- Điểm bài KTGK (2 bài) : chiếm 20 % điểm tổng kết
- Điểm kiểm tra cuối kỳ (điểm CK): chiếm 60% điểm tổng kết

Các thông tin còn lại trong bảng điểm được trích từ hồ sơ lý lịch sinh viên.

Viết chương trình (không phải viết hệ thống tùy chọn menu):

- Xuất ra Danh sách sinh viên gồm các thông tin : mã sinh viên, họ tên sinh viên, giới tính, quê quán, lớp

- Tính điểm tổng kết và xuất ra bảng điểm chi tiết môn học gồm các thông tin : Mã sinh viên, Điểm bài lab, điểm KTGK, Điểm CK, Điểm tổng kết.
- Xuất bảng điểm chính thức gồm các thông tin về sinh viên và điểm tổng kết môn học.

- Danh sách sinh viên cho trong bảng sau :

Danh sach sinh vien mon LTCT:

| :Ma NU   | :Ho          | Ten   | :GT  | :NS   | :Que quan   | :Lop   |
|----------|--------------|-------|------|-------|-------------|--------|
| :1512967 | :Trieu       | Minh  | :Nu  | :1997 | :Ninh Thuan | :CTK39 |
| :1510279 | :Nguyen Van  | Tan   | :Nam | :1997 | :Da Lat     | :CTK39 |
| :1512555 | :Tran Tuan   | Ngoc  | :Nam | :1996 | :Khanh Hoa  | :CTK39 |
| :1412120 | :Vo Thi      | Hoa   | :Nu  | :1996 | :Binh Dinh  | :CTK38 |
| :1313320 | :Le Ngoc     | Minh  | :Nam | :1995 | :Can Tho    | :CTK37 |
| :1510214 | :Dinh Thi    | Ngoc  | :Nu  | :1997 | :Da Lat     | :CTK39 |
| :1512128 | :Ta Van      | Ton   | :Nam | :1997 | :Binh Dinh  | :CTK39 |
| :1512868 | :Doan        | Du    | :Nam | :1996 | :Da Lat     | :CTK39 |
| :1512887 | :Vuong Ngoc  | Yen   | :Nu  | :1997 | :Da Nang    | :CTK39 |
| :1514205 | :Vuong Trung | Duong | :Nam | :1995 | :Phu Yen    | :CTK39 |
| :1510192 | :Nhac Linh   | San   | :Nu  | :1997 | :Da Lant    | :CTK39 |
| :1312890 | :Hoang       | Dung  | :Nu  | :1995 | :Da Nang    | :CTK37 |
| :1444405 | :Truong Vo   | Ky    | :Nam | :1996 | :Binh Dinh  | :CTK38 |
| :1512988 | :Vi Tieu     | Bao   | :Nam | :1996 | :Da Lat     | :CTK39 |

So luong sinh vien trong danh sach : n = 14\_

- Bảng điểm chi tiết môn học cho trong bảng sau:

Bang diem chi tiet mon LTCT:

| :Ma NU   | :Diem Lab | :Diem KTGK | :Diem Cuoi Ky | :Diem TK |
|----------|-----------|------------|---------------|----------|
| :1512967 | :9.0      | :8.0       | :6.0          | :7.0     |
| :1510279 | :10.0     | :6.0       | :5.0          | :6.2     |
| :1512555 | :8.0      | :5.0       | :7.0          | :6.8     |
| :1412120 | :9.5      | :9.0       | :8.0          | :8.5     |
| :1313320 | :8.0      | :8.0       | :4.0          | :5.6     |
| :1510214 | :8.5      | :8.0       | :6.0          | :6.9     |
| :1512128 | :9.0      | :9.0       | :9.0          | :9.0     |
| :1512868 | :10.0     | :8.5       | :8.5          | :8.8     |
| :1512887 | :7.0      | :7.0       | :7.0          | :7.0     |
| :1514205 | :8.0      | :8.0       | :8.5          | :8.3     |
| :1510192 | :10.0     | :9.0       | :9.0          | :9.2     |
| :1312890 | :9.0      | :9.0       | :8.0          | :8.4     |
| :1444405 | :7.0      | :5.0       | :5.0          | :5.4     |
| :1512988 | :8.5      | :8.0       | :8.0          | :8.1     |

So luong sinh vien trong danh sach : n = 14\_

- Hai bảng trên có cùng thứ tự trong cột mã sinh viên.

Chương trình tổ chức theo thư viện hàm (và không có hệ thống menu.)

Bước 1. Tạo Project rỗng mới đặt tên **Lab08\_D\_Bai3**

**Bước 2.** Tạo cấu trúc cho chương trình như đã hướng dẫn trong **mục 1 lab 4** (từ 1-7 để có cấu trúc nội dung tối thiểu chạy được chương trình, dạng không có hệ thống menu).

Tức là ta có kết quả chương trình ở bước này như sau :

- Tập tin **thuvien.h** : Rỗng (chỉ có các chú thích về cấu trúc văn bản)

- Tập tin **program.cpp** có nội dung như sau :

```
#include <iostream>
#include <conio.h>
using namespace std;
#include "thuvien.h"
void ChayChuongTrinh();
int main()
{
    ChayChuongTrinh();
    return 1;
}
void ChayChuongTrinh()
{
    _getch();
}
```

Nhấn Ctrl + F5 để chạy chương trình, sửa lỗi nếu có.

Bước 3:

Bước này ta bổ sung thêm định nghĩa hằng, kiểu dữ liệu mới,

- Trong tập tin **thuvien.h** :

+ Bổ sung định nghĩa hằng, kiểu dữ liệu mới :

```
//Định nghĩa hằng
#define MAX 100//kích thước khai mạng cấu trúc
#define NGANGDOI '=' //ke ngang đôi khi xuất dữ liệu
#define NGANGDON '-' //Ke ngang đơn khi xuất dữ liệu
```

```
//Định nghĩa kiểu dữ liệu mới : các kiểu cấu trúc
```

```
//Kiểu Sinh Viên
```

```
struct SinhVien
{
    char maSV[8];
    char hoLot[15];
    char ten[8];
    char gioiTinh[4];
    unsigned int namSinh;
    char queQuan[15];
    char lop[6];
};
```

```
struct BangDiem_ChiTiet
```

```
{
    char maSV[8];
    double diemLab;
    double baiKTGK;
    double baiKTCK;
    double diemTK;
};
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```
//Khai báo nguyên mẫu
```

```
//Định nghĩa các hàm
```

**Bước 4 :** Bổ sung các hàm nhập xuất dữ liệu và kiểm tra kết quả xuất danh sách sinh viên

4.1 Bổ sung các thư viện cần thiết :

- Trong tập tin **program.cpp** :

Bổ sung thêm thư viện **<string.h>**, **<iomanip>**

4.2 Định nghĩa các hàm nhập dữ liệu :

```
//=====//
///////// Tao Danh sach sinh vien ///////////////////////////////////////////
//=====//
//Chen 1 sinh vien
void Chen_SV(char maSV[8], char hoLot[15], char ten[7], char gioiTinh[3], unsigned int namSinh,
             char queQuan[15], char lop[7], SinhVien a[MAX], int &n)
{
    if (n < MAX)
    {
        strcpy_s(a[n].maSV, 8, maSV);
        strcpy_s(a[n].hoLot, 15, hoLot);
        strcpy_s(a[n].ten, 8, ten);
        strcpy_s(a[n].gioiTinh, 4, gioiTinh);
        a[n].namSinh = namSinh;
        strcpy_s(a[n].queQuan, 15, queQuan);
        strcpy_s(a[n].lop, 6, lop);

        n++;
    }
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

//Tao Danh Sach sinh vien

```
void TaoDanhSachSinhVien(SinhVien a[MAX], int &n)
{
    n = 0;
    Chen_SV("1512967", "Trieu", "Minh", "Nu", 1997, "Ninh Thuan", "CTK39", a, n);
    Chen_SV("1510279", "Nguyen Van", "Tan", "Nam", 1997, "Da Lat", "CTK39", a, n);
    Chen_SV("1512555", "Tran Tuan", "Ngoc", "Nam", 1996, "Khanh Hoa", "CTK39", a, n);
    Chen_SV("1412120", "Vo Thi", "Hoa", "Nu", 1996, "Binh Dinh", "CTK38", a, n);
    Chen_SV("1313320", "Le Ngoc", "Minh", "Nam", 1995, "Can Tho", "CTK37", a, n);
    Chen_SV("1510214", "Dinh Thi", "Ngoc", "Nu", 1997, "Da Lat", "CTK39", a, n);
    Chen_SV("1512128", "Ta Van", "Ton", "Nam", 1997, "Binh Dinh", "CTK39", a, n);
    Chen_SV("1512868", "Doan", "Du", "Nam", 1996, "Da Lat", "CTK39", a, n);
    Chen_SV("1512887", "Vuong Ngoc", "Yen", "Nu", 1997, "Da Nang", "CTK39", a, n);
    Chen_SV("1514205", "Vuong Trung", "Duong", "Nam", 1995, "Phu Yen", "CTK39", a, n);
    Chen_SV("1510192", "Nhac Linh", "San", "Nu", 1997, "Da Lant", "CTK39", a, n);
    Chen_SV("1312890", "Hoang", "Dung", "Nu", 1995, "Da Nang", "CTK37", a, n);
    Chen_SV("1444405", "Truong Vo", "Ky", "Nam", 1996, "Binh Dinh", "CTK38", a, n);
    Chen_SV("1512988", "Vi Tieu", "Bao", "Nam", 1996, "Da Lat", "CTK39", a, n);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

4.3 Định nghĩa các hàm xuất dữ liệu

```
//=====//
//////////  Xuat Danh sach sinh vien  //////////
//=====//
```

```
//Xuat ke ngang doi
void XuatKeNgangDoi()
```

```
{
    int i;
    cout << "\n";
    cout << setiosflags(ios::left)
        << ':';
    for (i = 1; i <= 66; i++)
        cout << NGANGDOI;
    cout << ':';
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```
//Xuat ke ngang don
void XuatKeNgangDon()
```

```
{
    int i;
    cout << "\n";
    cout << setiosflags(ios::left)
        << ':';
    for (i = 1; i <= 66; i++)
        cout << NGANGDON;
    cout << ':';
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```
//Xuat tieu de
void XuatTieuDe()
```

```
{
    XuatKeNgangDoi();
    cout << endl;
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << "Ma NV"
        << ':'
        << setw(15) << "Ho"
        << setw(7) << "Ten"
        << ':'
        << setw(5) << "GT"
        << ':'
        << setw(5) << "NS"
        << ':'
        << setw(15) << "Que quan"
        << ':'
        << setw(6) << "Lop"
        << ':';
    XuatKeNgangDoi();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```
//Xuat 1 sinh vien
```



```

void Xuat_1SV(SinhVien p)
{
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << p.maSV
        << ':'
        << setw(15) << p.hoLot
        << setw(7) << p.ten
        << ':'
        << setw(5) << p.gioiTinh
        << ':'
        << setw(5) << p.namSinh
        << ':'
        << setw(15) << p.queQuan
        << ':'
        << setw(6) << p.lop
        << ':';
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

//Xuat DS sinh vien

```

void Xuat_DSSV(SinhVien a[MAX], int n)
{
    int i;
    XuatTieuDe();
    for (i = 0; i < n; i++)
    {
        cout << endl;
        Xuat_1SV(a[i]);
        if ((i + 1) % 5 == 0)
            XuatKeNgangDon();
    }
    XuatKeNgangDoi();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

4.4 Khai báo nguyên mẫu các hàm nhập, xuất :

```

void Chen_SV(char maSV[8], char hoLot[15], char ten[7], char gioiTinh[3], unsigned int namSinh,
             char queQuan[15], char lop[7], SinhVien a[MAX], int &n);
void TaoDanhSachSinhVien(SinhVien a[MAX], int &n);
void XuatKeNgangDoi();
void XuatKeNgangDon();
void XuatTieuDe();
void Xuat_1SV(SinhVien p);
void Xuat_DSSV(SinhVien a[MAX], int n);

```

4.5 Kiểm tra chức năng nhập, xuất danh sách sinh viên

- Trong tập tin *program.cpp*, ta cập nhật hàm *ChayChuongTrinh* :

```

void ChayChuongTrinh()
{
    int n;
    SinhVien a[MAX];

    system("CLS");
    TaoDanhSachSinhVien(a, n);
}

```

```

cout << "\nDanh sach sinh vien mon LTCT:\n";
Xuat_DSSV(a, n);
cout << "\nSo luong sinh vien trong danh sach : n = " << n;
_getch();

}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.  
Kiểm tra các chức năng tạo và xuất danh sách sinh viên.

Bước 5: Tạo bảng điểm chi tiết và tính điểm tổng kết môn học

- Trong thuvien.h, ta bổ sung các hàm sau :

5.1 Tạo bảng điểm chi tiết :

```

//=====
//
//Tao bang diem chi tiet
//=====

//Chen diem 1 sinh vien
void Chen_Diem(char maSV[8], double diemLab, double baiKTGK, double baiKTCK, BangDiem_ChiTiet
bd[MAX], int &n)
{
    if (n < MAX)
    {
        strcpy_s(bd[n].maSV, 8, maSV);
        bd[n].diemLab = diemLab;
        bd[n].baiKTGK = baiKTGK;
        bd[n].baiKTCK = baiKTCK;
        bd[n].diemTK = 0; //khởi tạo
        n++;
    }
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```

//Tao bang diem chi tiet : chua tinh diem tong ket
void TaoBangDiem_ChiTiet(BangDiem_ChiTiet bd[MAX], int &n)
{
    n = 0;
    Chen_Diem("1512967", 9, 8, 6, bd, n);
    Chen_Diem("1510279", 10, 6, 5, bd, n);
    Chen_Diem("1512555", 8, 5, 7, bd, n);
    Chen_Diem("1412120", 9.5, 9, 8, bd, n);
    Chen_Diem("1313320", 8, 8, 4, bd, n);
    Chen_Diem("1510214", 8.5, 8, 6, bd, n);
    Chen_Diem("1512128", 9, 9, 9, bd, n);
    Chen_Diem("1512868", 10, 8.5, 8.5, bd, n);
    Chen_Diem("1512887", 7, 7, 7, bd, n);
    Chen_Diem("1514205", 8, 8, 8.5, bd, n);
    Chen_Diem("1510192", 10, 9, 9, bd, n);
    Chen_Diem("1312890", 9, 9, 8, bd, n);
    Chen_Diem("1444405", 7, 5, 5, bd, n);
    Chen_Diem("1512988", 8.5, 8, 8, bd, n);
}

```

```
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

//Tinh diem tong ket mon hoc

```
void TinhDiemTongKet(BangDiem_ChiTiet bd[MAX], int &n)
{
    int i;
    double tam;
    for (i = 0; i < n; i++)
    {
        tam = bd[i].diemLab * 0.2 + bd[i].baiKTGK * 0.2 + bd[i].baiKTCK * 0.6;
        bd[i].diemTK = tam;
    }
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

5.2 Các hàm xuất bange điểm chi tiết:

```
//=====
//          Xuat bang diem chi tiet
//=====
```

//Xuat tieu de bang diem chi tiet

```
void XuatTieuDe_bd()
{
    XuatKeNgangDoi();
    cout << endl;
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << "Ma NV"
        << ':'
        << setw(13) << "Diem Lab"
        << ':'
        << setw(13) << "Diem KTGK"
        << ':'
        << setw(13) << "Diem Cuoi Ky"
        << ':'
        << setw(15) << "Diem TK"
        << ':'
    XuatKeNgangDoi();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

//Xuat diem 1 sinh vien

```
void Xuat_Diem_1SV(BangDiem_ChiTiet p)
{
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << p.maSV
        << ':'
        << setw(13) << setiosflags(ios::fixed) << setprecision(1) << p.diemLab
        << ':'
        << setw(13) << setiosflags(ios::fixed) << setprecision(1) << p.baiKTGK
        << ':'
        << setw(13) << setiosflags(ios::fixed) << setprecision(1) << p.baiKTCK
}
```

```

        << '!';
        << setw(15) << setiosflags(ios::fixed) << setprecision(1) << p.diemTK
        << '!';
    }

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

```

//Xuat bang diem chi tiet
void Xuat_BangDiem_ChiTiet(BangDiem_ChiTiet bd[MAX], int n)
{
    int i;
    XuatTieuDe_bd();
    for (i = 0; i < n; i++)
    {
        cout << endl;
        Xuat_Diem_1SV(bd[i]);
        if ((i + 1) % 5 == 0)
            XuatKeNgangDon();
    }
    XuatKeNgangDoi();
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

5.3 Khai báo nguyên mẫu :

```

void Chen_Diem(char maSV[8], double diemLab, double baiKTGK, double baiKTCK, BangDiem_ChiTiet
bd[MAX], int &n);
void TaoBangDiem_ChiTiet(BangDiem_ChiTiet bd[MAX], int &n);
void TinhDiemTongKet(BangDiem_ChiTiet bd[MAX], int &n);
void XuatTieuDe_bd();
void Xuat_Diem_1SV(BangDiem_ChiTiet p);
void Xuat_BangDiem_ChiTiet(BangDiem_ChiTiet bd[MAX], int n);

```

5.4 Kiểm tra chức năng tạo và xuất bảng điểm chi tiết :

- Trong tập tin **program.cpp**, ta cập nhật lại hàm **ChayChuongTrinh** :

```

void ChayChuongTrinh()
{
    int n;
    SinhVien a[MAX];
    BangDiem_ChiTiet bd[MAX];

    system("CLS");
    TaoDanhSachSinhVien(a, n);
    cout << "\nDanh sach sinh vien mon LTCT:\n";
    Xuat_DSSV(a, n);
    cout << "\nSo luong sinh vien trong danh sach : n = " << n;
    _getch();

    system("CLS");
    TaoBangDiem_ChiTiet(bd, n);
    TinhDiemTongKet(bd, n);
    cout << "\nBang diem chi tiet mon LTCT:\n ";
    Xuat_BangDiem_ChiTiet(bd, n);
    cout << "\nSo luong sinh vien trong danh sach : n = " << n;
}

```

```
_getch();
```

```
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra các chức năng tạo và xuất bảng điểm chi tiết.

Bước 6: Xuất bảng điểm chính thức là bảng gồm các thông tin sinh viên và điểm tổng kết.

6.1 Hàm xuất bảng điểm chính thức:

```
//=====
//Xuất bang diem chinh thuc
//=====

void XuatBangDiem_ChinhThuc(SinhVien a[MAX], BangDiem_ChiTiet bd[MAX], int n)
{
    //xuat tieu de
    cout << endl << "=====:\n";
    cout << setiosflags(ios::left)
        << ':'
        << setw(8) << "Ma SV"
        << ':'
        << setw(15) << "Ho"
        << setw(7) << "Ten"
        << ':'
        << setw(5) << "GT"
        << ':'
        << setw(5) << "NS"
        << ':'
        << setw(15) << "Que quan"
        << ':'
        << setw(6) << "Lop"
        << ':'
        << setw(8) << "Diem TK"
        << ':'
    cout << endl << "=====:";

    int i;
    for (i = 0; i < n; i++)
    {
        cout << endl;
        cout << setiosflags(ios::left)
            << ':'
            << setw(8) << a[i].maSV
            << ':'
            << setw(15) << a[i].hoLot
            << setw(7) << a[i].ten
            << ':'
            << setw(5) << a[i].gioiTinh
            << ':'
            << setw(5) << a[i].namSinh
            << ':'
            << setw(15) << a[i].queQuan
            << ':'
            << setw(6) << a[i].lop
            << ':'
    }
```

```

                << setw(8) << setiosflags(ios::fixed) << setprecision(1) << bd[i].diemTK
                << '!';
            if ((i + 1) % 5 == 0)
                cout << endl << ":-:-----:~";
        }
        cout << endl << "=====:";
    }
}

```

6.2 Khai báo nguyên mẫu:

```
void XuatBangDiem_ChinhThuc(SinhVien a[MAX], BangDiem_ChiTiet bd[MAX], int n);
```

6.3. Kiểm tra chức năng xuất bảng điểm chính thức:

- Trong tập tin *program.cpp*, ta cập nhật lại hàm *ChayChuongTrinh* :

```

void ChayChuongTrinh()
{
    int n;
    SinhVien a[MAX];
    BangDiem_ChiTiet bd[MAX];

    system("CLS");
    TaoDanhSachSinhVien(a, n);
    cout << "\nDanh sach sinh vien mon LTCT:\n";
    Xuat_DSSV(a, n);
    cout << "\nSo luong sinh vien trong danh sach : n = " << n;
    _getch();

    system("CLS");
    TaoBangDiem_ChiTiet(bd, n);
    TinhDiemTongKet(bd, n);
    cout << "\nBang diem chi tiet mon LTCT:\n ";
    Xuat_BangDiem_ChiTiet(bd, n);
    cout << "\nSo luong sinh vien trong danh sach : n = " << n;
    _getch();

    system("CLS");
    cout << "\nBang diem chinh thuc mon LTCT:\n ";
    XuatBangDiem_ChinhThuc(a, bd, n);
    cout << "\nSo luong sinh vien trong danh sach : n = " << n;
    cout << endl;
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra chức năng xuất bảng điểm chính thức.

Kiểm tra thực hiện các yêu cầu banif toán – kết thúc chương trình.

Bài 4:

Đa thức một biến bậc n, hệ số thực có dạng:

$$F(x) = c_0 + c_1x^1 + \dots + c_nx^n ; c_i \neq 0, \forall i$$

Có thể biểu diễn bởi một mảng các đơn thức có dạng  $c_i x^i$ ,  $c_i \neq 0$ .

Mỗi đơn thức là một cấu trúc gồm 2 trường: **Hệ số và số mũ**.

Viết chương trình tùy chọn thực hiện các chức năng dưới đây:

0. Thoát khỏi chương trình
1. Nhập đa thức
2. Xuất đa thức
3. Công 2 đa thức
4. Trừ đa thức
5. Nhân 2 đa thức
6. Tính giá trị của đa thức tại x

Tạo project rỗng mới, đặt tên **Lab08\_D\_Bai4**.

Tổ chức chương trình tùy chọn menu như bài 1.

- Tham khảo tập tin **thuvien.h** sau đây :

```
//Định nghĩa hằng
#define MAX 100
//Định nghĩa kiểu dữ liệu
//Kiểu đơn thức
struct DonThuc
{
    double heSo;
    int mu;
};

//Kiểu đa thức : mảng các đơn thức
typedef DonThuc DaThuc[MAX];

//Khai báo nguyên mẫu
void RutGon_DT(DaThuc A, int &n);
void Khu_HS0(DaThuc A, int &n);
void SapTangTheoMu(DaThuc A, int n);
void DinhDang_DT(DaThuc A, int &n);

void NhapDaThuc(DaThuc A, int &n, char kt);
void XuatDaThuc(DaThuc A, int n);

void Tong2DaThuc(DaThuc A, int m, DaThuc B, int n, DaThuc C, int &k);
void Hieu2DaThuc(DaThuc A, int m, DaThuc B, int n, DaThuc C, int &k);
void Nhan_DonThuc(DaThuc A, int m, DonThuc p, DaThuc C, int &k);
void NhanDaThuc(DaThuc A, int m, DaThuc B, int n, DaThuc C, int &k);
double x_Mu_m(double x, int m);
double A_x(DaThuc A, int m, double x);

//Định nghĩa các hàm

//:=====
//:Định dạng đa thức : Rut gọn - Khu hệ số 0 - Sắp tang theo số mũ :
//:=====
//Rut gọn đa thức : đơn các đơn thức có số mũ bằng nhau lại với nhau
void RutGon_DT(DaThuc A, int &n)
{
    DaThuc B; //Biến trung gian
    int i, //duyet A
        j; //duyet B
    int h = 0, //kich thước khởi tạo của B
        kq;
    for (i = 0; i < n; i++)
```

```

{
    kq = 0;
    for (j = 0; j < h; j++)
        if (A[i].mu == B[j].mu)
        {
            kq = 1;
            break;
        }
    if (kq == 1)
        B[j].heSo += A[i].heSo;
    else
        B[h++] = A[i];
}
n = h;
for (j = 0; j < h; j++) //Gan B cho A
    A[j] = B[j];
}

//Loai he so 0
void Khu_HS0(DaThuc A, int &n)
{
    int i,
        h = 0;
    DaThuc B;
    for (i = 0; i < n; i++)
        if (A[i].heSo != 0)
            B[h++] = A[i];
    n = h;
    for (i = 0; i < n; i++)
        A[i] = B[i];
}

//Sap tang theo so mu
void SapTangTheoMu(DaThuc A, int n)
{
    int i, j;
    DonThuc t;
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (A[i].mu > A[j].mu)
            {
                t = A[i];
                A[i] = A[j];
                A[j] = t;
            }
}

void DinhDang_DT(DaThuc A, int &n)
{
    if (n == 0)
    {
        cout << "\nDa thuc rong!";
        _getch();
        return;
    }
}

```



```

        RutGon_DT(A, n);
        Khu_HSO(A, n);
        SapTangTheoMu(A, n);
    }

//:=====
//:Nhap, Xuat da thuc : da thuc da duoc Dinh dang      :
//:=====

//Nhap da thuc
void NhapDaThuc(DaThuc A, int &n, char kt)
{
    int i;
    cout << "\nNhap kích thước mảng : ";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cout << "\nĐón thực " << kt << "[" << i << "] : ";
        cout << "\nNhap he so != 0 : he so = ";
        cin >> A[i].heSo;
        cout << "\nNhap so mu : So mu = ";
        cin >> A[i].mu;
    }
    DinhDang_DT(A, n);
}

//Xuat Da thuc : mu bieu dien bang ^
void XuatDaThuc(DaThuc A, int n)
{
    if (n == 0)
    {
        cout << "\nDa thuc rong!";
        _getch();
        return;
    }
    int i = 0;
    //xu ly don tu dau tien, so mu co the == 0
    if (A[i].mu == 0) //so mu = 0
        cout << A[i].heSo;
    else // so mu > 0 : >= 1
    {
        if (A[i].mu == 1) // so mu = 1
        {
            if (A[i].heSo == 1) //he so == 1
                cout << "x ";
            else // he so != 1
                if (A[i].heSo == -1)
                    cout << "-x ";
            else
                cout << A[i].heSo << "x ";
        }
        else //so mu > 1
        {
            if (A[i].heSo == 1) //he so == 1
                cout << "x^" << A[i].mu << " ";
        }
    }
}

```

```

        else // he so !=1
        if (A[i].heSo == -1)
            cout << "-x^" << A[i].mu << " ";
        else
            cout << A[i].heSo << "x^" << A[i].mu << " ";
    }
}
//Xu ly cac don thuc tiep theo
i++;
while (i < n)
{
    //so mu > 0
    if (A[i].mu == 1) // so mu = 1
    {
        if (A[i].heSo == 1) //he so == 1
            cout << "+x ";
        else // he so !=1
        if (A[i].heSo == -1)
            cout << "-x ";
        else
        if (A[i].heSo > 0)
            cout << "+" << A[i].heSo << "x ";
        else
            cout << A[i].heSo << "x ";
    }
    else //so mu > 1
    {
        if (A[i].heSo == 1) //he so == 1
            cout << "+x^" << A[i].mu << " ";
        else // he so !=1
        if (A[i].heSo == -1)
            cout << "-x^" << A[i].mu << " ";
        else
        if (A[i].heSo > 0)
            cout << "+" << A[i].heSo << "x^" << A[i].mu << " ";
        else //(he so < 0)
            cout << A[i].heSo << "x^" << A[i].mu << " ";
    }
    i++;
}
}

//:=====
//:Cac phep toan da thuc      :
//:=====
//C = A + B
void Tong2DaThuc(DaThuc A, int m, DaThuc B, int n, DaThuc C, int &k)
{
    int i = 0, j = 0;
    k = 0;
    DonThuc x;

    while (i < m && j < n)
    {
        if (A[i].mu < B[j].mu)

```

```

        {
            C[k++] = A[i];
            i++;
        }
        else
        if (B[j].mu < A[i].mu)
        {
            C[k++] = B[j];
            j++;
        }
        else //2 mu bang nhau
        {
            x.heSo = A[i].heSo + B[j].heSo;
            x.mu = A[i].mu;
            if (x.heSo != 0) // khu he so 0
                C[k++] = x;
            i++;
            j++;
        }
    }

    while (i < m)
    {
        C[k++] = A[i];
        i++;
    }

    while (j < n)
    {
        C[k++] = B[j];
        j++;
    }
    DinhDang_DT(C, k);
}

//C = A - B
void Hieu2DaThuc(DaThuc A, int m, DaThuc B, int n, DaThuc C, int &k)
{
    int i = 0, j = 0;
    k = 0;
    DonThuc x;

    while (i < m && j < n)
    {
        if (A[i].mu < B[j].mu)
        {
            C[k++] = A[i];
            i++;
        }
        else
        if (B[j].mu < A[i].mu)
        {
            x.mu = B[j].mu;
            x.heSo = -B[j].heSo;
            C[k++] = x;

```

```

        j++;
    }
    else //2 mu bang nhau
    {
        x.heSo = A[i].heSo - B[j].heSo;
        x.mu = A[i].mu;
        if (x.heSo != 0) // khu he so 0
            C[k++] = x;
        i++;
        j++;
    }
}

while (i < m)
{
    C[k++] = A[i];
    i++;
}

while (j < n)
{
    x.mu = B[j].mu;
    x.heSo = -B[j].heSo;
    C[k++] = x;
    j++;
}
DinhDang_DT(C, k);
}

//Nhan mot da thuc voi mot don thuc
void Nhan_DonThuc(DaThuc A, int m, DonThuc p, DaThuc C, int &k)
{
    int i;
    DonThuc x;
    k = 0;
    for (i = 0; i < m; i++)
    {
        x.mu = A[i].mu + p.mu;
        x.heSo = A[i].heSo * p.heSo;
        C[k++] = x;
    }
    DinhDang_DT(C, k);
}

//C = AB
void NhanDaThuc(DaThuc A, int m, DaThuc B, int n, DaThuc C, int &k)
{
    DaThuc T1, T2;
    int i, j, h1, h2;
    Nhan_DonThuc(B, n, A[0], C, k);

    for (i = 1; i < m; i++)
    {
        Nhan_DonThuc(B, n, A[i], T1, h1);

```

```

        Tong2DaThuc(C, k, T1, h1, T2, h2);
        for (j = 0; j < h2; j++)
            C[j] = T2[j];
        k = h2;
    }
    DinhDang_DT(C, k);
}

```

```

double x_Mu_m(double x, int m)
{
    double t = 1;
    int i;
    for (i = 1; i <= m; i++)
        t *= x;
    return t;
}

```

//Tinh A(x)

```

double A_x(DaThuc A, int m, double x)
{
    double v = 0, t;
    int i;
    for (i = 0; i < m; i++)
    {
        t = A[i].heSo * x_Mu_m(x, A[i].mu);
        v += t;
    }
    return v;
}

```

- Để khỏi mất thời gian nhập dữ liệu, có thể khởi đầu các đa thức A, B trong hàm ChayChuongTrinh() của tập tin **program.cpp**. Trong quá trình thực hiện chương trình, muốn thay đổi bộ dữ liệu mới có thể chọn thực hiện chức năng 1 để nhập đa thức từ bàn phím.

```

void ChayChuongTrinh()
{
    int soMenu = 6,
        menu;

    int m = 3, n = 4; //khởi đầu m, n

    //khởi đầu A, B
    DaThuc A = {
        {1,1},
        {4,2},
        {-5,6}
    },
    B = {
        {1, 0 },
        {-4, 2},
        {3, 3 },
        {6, 4 }
    };
}

```

```

do
{
    system("CLS");
    menu = ChonMenu(soMenu);
    XuLyMenu(menu, A,m, B, n);
} while (menu > 0);
}

```

## E. Bài tập bắt buộc

**Tất cả các bài tập phải được tổ chức dưới dạng thư viện hàm và có menu chức năng.**

### Bài 1 : (Quản lý điểm sinh viên)

Bảng điểm sinh viên chứa các thông tin :

- Mã sinh viên
- Họ và Tên sinh viên
- Năm sinh
- Lớp
- Điểm Trung Bình

Viết chương trình tùy chọn menu thực hiện các chức năng :

0. Thoát khỏi chương trình
1. Tạo bảng điểm sinh viên
2. Xem bảng điểm sinh viên.
3. Xuất bảng điểm sinh viên theo từng lớp
4. Sắp bảng điểm sinh viên giảm dần theo điểm trung bình
5. Sắp xếp và xuất DSSV tăng dần theo tên, nếu trùng tên, sắp tăng theo họ và tên lót.
6. Tìm và xuất thông tin của sinh viên có tên cho trước.
7. Tìm và xuất thông tin của các sinh viên có điểm trung bình cao nhất.
8. Xếp loại học lực của sinh viên dựa vào điểm trung bình. Biết rằng:
  - Giỏi:  $8.5 \leq \text{Điểm trung bình} \leq 10$
  - Khá:  $7.0 \leq \text{Điểm trung bình} < 8.5$
  - Trung bình :  $5.5 \leq \text{Điểm trung bình} < 7.0$
  - Yếu:  $4.0 \leq \text{Điểm trung bình} < 5.5$
  - Kém:  $0.0 \leq \text{Điểm trung bình} < 4.0$

### Bài 2. Quản lý thuê bao điện thoại

Viết chương trình quản lý các khách hàng thuê bao điện thoại. Trong đó, với mỗi thuê bao, cần quản lý những thông tin sau:

- Mã số thuê bao: là một số nguyên dương.
- Họ và tên: là một chuỗi chứa tối đa 50 ký tự.
- Địa chỉ: là một chuỗi chứa tối đa 100 ký tự.
- Số điện thoại: là một chuỗi chứa 10 hoặc 11 chữ số và một dấu chấm (.) để phân cách mã vùng với số điện thoại. Ký tự đầu tiên là ký tự 0.
- Ngày hợp đồng: cho biết ngày, tháng, năm hợp đồng thuê bao.

Chương trình cho phép người dùng thực hiện những chức năng dưới đây:

0. Thoát khỏi chương trình
1. Nhập danh sách thuê bao
2. Xem danh sách thuê bao
3. Tìm số điện thoại khi biết tên
4. Tìm thông tin thuê bao khi biết số điện thoại
5. Xuất các thuê bao có cùng địa chỉ
6. Sắp xếp các thuê bao tăng dần theo mã vùng.

### Bài 3. Vector trong không gian 3 chiều

Viết chương trình thực hiện các phép toán trên vector trong không gian 3 chiều theo các yêu cầu dưới đây. Giả sử cho 2 vector có tọa độ như sau:  $\vec{u} = (x_u, y_u, z_u)$  và  $\vec{v} = (x_v, y_v, z_v)$

1. Tính độ dài vector  $\vec{u}$  theo công thức  $|\vec{u}| = \sqrt{x_u^2 + y_u^2 + z_u^2}$ .
2. Kiểm tra hai vector  $\vec{u}$  và  $\vec{v}$  có bằng nhau không.
3. Tính tổng của hai vector  $\vec{u}$  và  $\vec{v}$  theo công thức  $\vec{u} + \vec{v} = (x_u + x_v, y_u + y_v, z_u + z_v)$
4. Tính hiệu của hai vector  $\vec{u}$  và  $\vec{v}$  theo công thức  $\vec{u} - \vec{v} = (x_u - x_v, y_u - y_v, z_u - z_v)$
5. Tính tích của một số thực với một vector theo công thức  $k \cdot \vec{u} = (k \cdot x_u, k \cdot y_u, k \cdot z_u)$
6. Tính tích vô hướng của hai vector  $\vec{u}$  và  $\vec{v}$  theo công thức  $\vec{u} \cdot \vec{v} = x_u x_v + y_u y_v + z_u z_v$
7. Kiểm tra hai vector  $\vec{u}$  và  $\vec{v}$  có vuông góc với nhau hay không? Biết rằng, hai vector  $\vec{u}$  và  $\vec{v}$  vuông góc nếu tích vô hướng của chúng bằng 0.

## F. Bài tập làm thêm

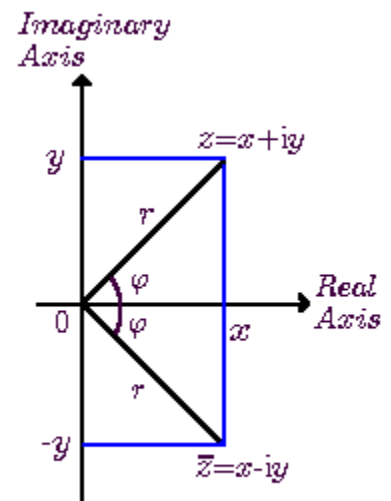
### Bài 1. Số phức

Số phức là số có dạng  $a + bi$ , trong đó  $a$  và  $b$  là các số thực,  $i$  là đơn vị ảo với  $i^2 = -1$ . Trong biểu thức này,  $a$  gọi là phần thực và  $b$  gọi là phần ảo của số phức. Số phức có thể được biểu diễn trên mặt phẳng phức với trục hoành là trục thực và trục tung là trục ảo. Do đó, một số phức  $a + bi$  có thể được xác định bằng một điểm có tọa độ  $(a, b)$ . Một số phức nếu có phần thực bằng 0 thì được gọi là **số thuần ảo**, nếu phần ảo bằng 0 thì trở thành số thực.

Hãy viết chương trình thực hiện các chức năng sau trên số phức:

- Tạo số phức bằng cách nhập phần thực và phần ảo từ bàn phím.
- Xuất số phức ra màn hình.
- Kiểm tra hai số phức có bằng nhau hay không.
- Tính độ lớn (magnitude) hay mô-đun của số phức  $z = a + bi$  theo công thức  $|z| = \sqrt{a^2 + b^2}$ .
- Tìm số phức liên hợp của số phức  $a + bi$ .
- Tìm số đối của số phức  $a + bi$ .
- Tìm số nghịch đảo của số phức (khác 0)  $a + bi$  theo công thức:  

$$\frac{a}{a^2 + b^2} - \frac{b}{a^2 + b^2} i$$
- Kiểm tra một số phức  $a + bi$  có phasê là số thuần ảo hay không.
- Cộng hai số phức theo công thức:  $(a+bi) + (c+di) = (a+c) + (b+d)i$ .
- Trừ số phức cho một số phức theo công thức:  $(a+bi) - (c+di) = (a-c) + (b-d)i$ .
- Nhân một số thực với một số phức.
- Nhân hai số phức theo công thức:  $(a+bi) * (c+di) = (ac-bd) + (ad+bc)i$ .
- Chia một số phức cho một số phức:



$$\frac{a+bi}{c+di} = \frac{(a+bi)(c-di)}{(c+di)(c-di)} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$$

- Tìm Argumen của số phức  $z = a + bi$ . Biết  $\text{Argumen}(z)$  chính là góc hợp bởi chiều dương của trục Ox và vector  $\overrightarrow{OM}$  với  $M(a,b)$  là tọa độ của  $z$  khi biểu diễn trên mặt phẳng tọa độ.
- Xuất số phức  $z = a + bi$  ở dạng lượng giác theo công thức:

$$z = r(\cos\varphi + i \sin\varphi) \text{ với } r = |z|, \varphi = \text{Argumen}(z)$$

### Bài 1. Quản lý sách

Viết chương trình quản lý một thư viện sách (gọi chung là tài liệu) đáp ứng các chức năng sau:

- Nhập một danh sách tài liệu.
- Xuất danh sách tài liệu ra màn hình.
- Tính tổng giá tất cả các tài liệu
- Tìm danh mục sách được xuất bản bởi **nhaXb** vào **namXb** cho trước.
- Tìm những bài báo khoa học có sự tham gia của tác giả **tacGia** cho trước.
- Thống kê số lượng tài liệu theo mỗi loại.
- Xem thông tin những tài liệu có giá đắt nhất.
- Liệt kê các tài liệu theo từng năm xuất bản.
- Tìm tài liệu có nhiều tác giả nhất.
- Xem thông tin tài liệu theo mã số tài liệu (**maTL**) cho trước.
- Sắp xếp các tài liệu tăng dần theo tựa đề.

Biết rằng, với mỗi tài liệu, cần quản lý các thông tin sau:

- Mã tài liệu: là một chuỗi chứa tối đa 10 ký tự.
- Tựa đề
- Loại tài liệu: thuộc một trong các loại sau: Sách, Báo khoa học, Tạp chí, Luận văn.
- Năm xuất bản
- Tác giả: lưu danh sách tác giả của tài liệu (phân tách nhau bởi dấu phẩy), trường này không dùng cho Tạp chí.
- Nhà xuất bản: không dùng cho Luận văn.
- Giá (mua về hoặc đền bù nếu mất): tính theo VNĐ.

### Bài 3. Quản lý độc giả

Viết chương trình quản lý độc giả trong một thư viện theo các yêu cầu sau:

1. Nhập một danh sách độc giả.
2. Xuất danh sách độc giả (DSDG) tăng dần theo tên, nếu trùng tên, sắp tăng theo họ.
3. Sắp xếp và xuất DSDG giảm dần theo ngày cấp thẻ.
4. Tìm những độc giả hiện tại không mượn cuốn sách nào.
5. Thống kê tỷ lệ (phần trăm) số lượng độc giả theo từng lứa tuổi như sau:
  - a. Người già: Trên 55 tuổi
  - b. Trung niên: Từ 40 tuổi đến dưới 55 tuổi
  - c. Thanh niên: Từ 20 tuổi đến dưới 40 tuổi
  - d. Thiếu niên: Từ 10 tuổi đến dưới 20 tuổi
  - e. Trẻ em: Dưới 10 tuổi
6. Tìm thông tin độc giả khi biết tên.
7. Tìm thông tin độc giả khi biết cả họ và tên.
8. Liệt kê những độc giả đang mượn ít nhất 3 cuốn sách.



9. Xem thông tin những độc giả mới được cấp thẻ gần đây nhất.
10. Xem thông tin của độc giả cao tuổi nhất.
11. Đếm số lượng độc giả là trẻ em theo phân loại ở câu 5.
12. Tính độ tuổi trung bình của tất cả các độc giả trong thư viện.
13. Tính tổng số sách hiện đang được mượn bởi các độc giả.
14. Thống kê số lượng sách đang được mượn theo từng lứa tuổi (theo phân loại như ở câu 5).
15. Tìm những độc giả có họ (**hoDocGia** nhập từ bàn phím) cho trước.

Biết rằng, với mỗi độc giả, cần quản lý các thông tin sau:

- Mã độc giả: là một số nguyên dương
- Họ và tên lót của độc giả
- Tên độc giả
- Tuổi
- Ngày cấp thẻ độc giả
- Số sách đang mượn: là một số nguyên không âm, có giá trị từ 0 tới 5.

#### **Bài 4. Cửa hàng bách hóa**

Viết chương trình quản lý các mặt hàng trong một cửa hàng bách hóa theo các yêu cầu sau:

- Nhập một danh sách mặt hàng
- Xuất các mặt hàng ra màn hình theo thứ tự ABC của tên mặt hàng.
- Tính tổng giá trị của tất cả các mặt hàng.
- Cho biết các mặt hàng có giá trị lớn nhất trong cửa hàng.
- Xuất ra những mặt hàng đã hết hạn sử dụng.
- Sắp xếp và xuất các mặt hàng theo từng loại.
- Tìm những mặt hàng còn hạn sử dụng không quá 30 ngày.
- Liệt kê những mặt hàng đã bán hết (số lượng tồn bằng 0).
- Tính số lượng mặt hàng bánh và kẹo.
- Tìm mặt hàng bia có giá rẻ nhất.

Với mỗi mặt hàng, cần quản lý các thông tin sau:

- Tên mặt hàng
- Đơn giá (tính theo VNĐ)
- Số lượng
- Đơn vị tính (gói, hộp, thùng, kg, ...)
- Hạn sử dụng (kiểu ngày tháng)
- Loại mặt hàng (bánh, kẹo, đồ uống, sữa, bia, ...)