

## LAB 9. BIẾN ĐỘNG VÀ KIỂU CON TRỎ

THỜI LƯỢNG: 6 TIẾT

### A. Mục tiêu

- Giúp sinh viên hiểu rõ và thực hiện thuần thục các kỹ thuật xử lý trên con trỏ.
- Sau khi hoàn thành bài thực hành này, sinh viên cần:
  - Nắm vững các khái niệm và cách định nghĩa kiểu con trỏ, khai báo biến con trỏ.
  - Nắm vững các kỹ thuật xử lý cơ bản trên kiểu con trỏ.
  - Hiểu rõ cách cấp phát bộ nhớ, truy cập phần tử và thu hồi vùng nhớ cấp phát cho mảng động.
  - Cài đặt được các thao tác cơ bản trên mảng động, chuỗi ký tự động bằng con trỏ.
  - Biết cách sử dụng con trỏ kết hợp với cấu trúc.
  - Hiểu rõ cơ chế gọi hàm, truyền tham trị và truyền tham biến

### 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, 4: 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 14**

### C. Ôn tập lý thuyết

#### 1. Kiểu con trỏ và biến con trỏ

Cú pháp khai báo biến con trỏ: **KDL** \* **Tên\_biến\_con\_trỏ**;

Ví dụ: `int *px; double *pd; NhanVien *nv;`

Cú pháp định nghĩa kiểu con trỏ: **typedef KDL** \* **Tên\_kiểu\_con\_trỏ**

Ví dụ: `typedef int *IntPtr;`  
`typedef Node *NodePointer;`

#### 2. Các phép toán trên con trỏ

- Lấy nội dung tại địa chỉ mà con trỏ px trỏ tới: **\* px**
- Phép gán địa chỉ cho con trỏ cùng kiểu: **int \*px, x = 3; px = &x;**
- Phép gán hai con trỏ cùng kiểu: **int \*py; py = px;**
- Mọi con trỏ đều có thể nhận giá trị NULL: **py = NULL;**
- Cấp phát vùng nhớ cho biến con trỏ: **px = new KDL;**
- Thu hồi vùng nhớ cho biến con trỏ: **delete px;**

#### 3. Mảng động một chiều

Cấp phát động cho mảng một chiều thông qua con trỏ (hay dùng con trỏ để cài đặt mảng 1 chiều).

- Khai báo **KDL** \*tên\_biến\_mảng;
- Cấp phát vùng nhớ tên\_biến\_mảng = **new KDL** [ **Kích\_thước\_mảng** ];
- Thu hồi (giải phóng) vùng nhớ **delete** [ ]tên\_biến\_mảng;
- Truy cập đến phần tử trong mảng

Truy cập đến giá trị	a[i]	*(a+i)
Truy cập đến địa chỉ	&a[i]	a+i

#### 4. Mảng động hai chiều (ma trận)

Cấp phát động cho mảng hai chiều thông qua con trỏ

- Khai báo **KDL** \*tên\_biến\_mảng;
- Cấp phát vùng nhớ tên\_biến\_mảng = **new KDL** [ **Số\_dòng** \* **Số\_cột** ];
- Thu hồi (giải phóng) vùng nhớ **delete** [ ]tên\_biến\_mảng;
- Truy cập đến phần tử ở hàng i, cột j trong mảng theo chỉ số (ví dụ: a[i][j]) hoặc sử dụng con trỏ \*(a + i\*n + j).

#### 5. Chuỗi ký tự động

Cấp phát động cho chuỗi ký tự thông qua con trỏ

- Khai báo **char** \*tên\_biến\_chuỗi;
- Khởi tạo tên\_biến\_mảng = **NULL**;
- Cấp phát vùng nhớ tên\_biến\_mảng = **new char** [ **Số\_ký\_tự\_tối\_đa** ];
- Thu hồi (giải phóng) vùng nhớ **delete** [ ]tên\_biến\_mảng;
- Truy cập đến từng ký tự (phần tử) giống như mảng động một chiều.

#### 6. Con trỏ cấu trúc

Giả sử ta có một kiểu dữ liệu cấu trúc có tên là KCT.

- Khai báo **KCT** \*tên\_biến;
- Cấp phát vùng nhớ tên\_biến = **new KCT**;
- Thu hồi (giải phóng) vùng nhớ **delete** tên\_biến;
- Truy cập đến các thành phần (trường) của cấu trúc: tên\_biến -> Tên\_thành\_phần

Ví dụ:

PhanSo *p;	p->MauSo = 5;
p = new PhanSo;	cout << p->TuSo << ' ' << p->MauSo;
p->TuSo = 2;	delete p;

#### 7. Mảng động cấu trúc

- Khai báo **KCT** \*tên\_biến;
- Cấp phát vùng nhớ tên\_biến = **new KCT** [ **Kích\_thước** ];
- Thu hồi (giải phóng) vùng nhớ **delete** [ ] tên\_biến;

#### Chú ý:

- Theo cách khai báo này, mỗi phần tử của mảng là một cấu trúc, không phải con trỏ cấu trúc.

#### 8. Truyền tham số

##### a. Một số điểm cần lưu ý

- Muốn hàm trả về một giá trị là mảng hoặc xâu thì khai báo kiểu trả về là kiểu con trỏ.
- Muốn tham số thực giữ lại giá trị đã bị thay đổi (bên trong hàm) khi chương trình ra khỏi hàm thì phải dùng cách truyền tham biến. Nghĩa là đối số phải là con trỏ hoặc tham chiếu.

##### b. Cách truyền tham số

Tham số hình thức	Tham số thực (Đối số)	Ghi chú
Biến	Giá trị	Truyền tham trị
Con trỏ	Con trỏ Địa chỉ của biến Tên mảng một chiều Tên mảng hai chiều (phải ép kiểu) Tên xâu ký tự	Truyền tham biến hay truyền bằng biến
Tham chiếu	Giá trị	
Tên mảng	Tên mảng	
Tên xâu ký tự	Tên xâu ký tự	

## D. Hướng dẫn thực hành

### Bài 1. Mảng động 1 chiều

Viết chương trình tùy chọn thực hiện các chức năng sau đây trên mảng số nguyên :

0. Thoát khỏi chương trình
1. Nhập tự động mảng
2. Xem mảng
3. Tính giá trị nhỏ nhất
4. Đếm số đường chạy của mảng
5. Tính tổng các số nguyên tố trong mảng
6. Tìm vị trí cuối cùng x xuất hiện trong mảng, nếu có

Yêu cầu:

- Cài đặt mảng động
- Các thuật toán cài đặt dưới hình thức đệ quy

Tạo project riêng, đặt tên **MaSV\_Lab09\_D\_Bai1**.

Chương trình tổ chức theo thư viện hàm và có hệ thống menu, phát triển như bài 1 mục D lab 8.

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

```
#define TAB '\t'
typedef int *DayDong;

voidNhapTuDong(DayDong a, int n)
{
    int i;
    // Gieo số ngẫu nhiên đầu tiên
    srand((unsigned)time(NULL));
    // Duyệt qua từng phần tử từ vị trí 0 tới n-1
    for (i = 0; i < n; i++)
    {
        // Sinh một số ngẫu nhiên trong phạm vi
        // [-20..20) rồi gán cho phần tử thứ i
        *(a+i) = -20 + rand() % 40;
    }
}

voidXuatMang(DayDong a, int n)
{
    int i;
    for (i = 0; i < n; i++)
```

```
        cout << *(a+i) << TAB;
    }
//Tính min
int TinhMin(DayDong a, int n)
{
    int kq;
    if (n == 1)
        kq = *(a+0);
    else
    if (n > 1)
    if (TinhMin(a, n - 1) > *(a + n - 1))
        kq = *(a + n - 1);
    else
        kq = TinhMin(a, n - 1);
    return kq;
}

//Tính số dương chạy
int TinhSo_DC(DayDong a, int n)
{
    int kq;
    if (n == 1)
        kq = 1;
    else
    if (n > 1)
    if (*(a + n - 1) < *(a + n - 2))
        kq = TinhSo_DC(a, n - 1) + 1;
    else
        kq = TinhSo_DC(a, n - 1);
    return kq;
}

//Kiểm tra nguyên tố
int KiemTra_NT(int x)
{
    int i, m,
        kq;
    if (x < 2)
        kq = 0;
    else
    {
        m = (int)sqrt((double)x);
        kq = 1;
        for (i = 2; i <= m; i++)
            if (x % i == 0)
            {
                kq = 0;
                break;
            }
    }
    return kq;
}

//Tính tổng nguyên tố
```

```

int TinhTong_NT(DayDong a, int n)
{
    int kq;
    if (n == 1)
        if (KiemTra_NT(*a))
            kq = *a;
        else
            kq = 0;
    else
        if (n > 1)
            if (KiemTra_NT(*(a + n - 1)))
                kq = TinhTong_NT(a, n - 1) + *(a + n - 1);
            else
                kq = TinhTong_NT(a, n - 1);
    return kq;
}
//vi tri cuoi cung x xuất hiện
int Tim_Csccl(DayDong a, int n, int x)
{
    int kq;
    if (n == 1)
        if (*a == x)
            kq = 0;
        else
            kq = -1;
    else
        if (n > 1)
            if (*(a + n - 1) == x)
                kq = n - 1;
            else
                kq = Tim_Csccl(a, n - 1, x);
    return kq;
}

//=====

```

## Bài 2. Mảng động 2 chiều

Viết chương trình tùy chọn thực hiện các phép toán trên ma trận vuông các số nguyên :

0. Thoát khỏi chương trình
1. Nhập tự động ma trận
2. Xem mảng
3. Cộng ma trận
4. Trừ ma trận
5. Nhân ma trận

Yêu cầu:

- Cài đặt mảng động

Tạo project riêng, đặt tên **MaSV\_Lab09\_D\_Bai2**.

Chương trình tổ chức theo thư viện hàm và có hệ thống menu, phát triển như bài 1 mục D lab 8.

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

```

//Định nghĩa hàng
//Định nghĩa kiểu dữ liệu mới
typedef int *MaTranVuong;

```

```
//Khai bao nguyen mau cac ham
void NhapMaTran(MaTranVuong &a, int n);
void XuatMaTran(MaTranVuong a, int n);
/*
void TinhTong_2_MaTran(MaTranVuong a, MaTranVuong b, MaTranVuong c, int n);
void TinhHieu_2_MaTran(MaTranVuong a, MaTranVuong b, MaTranVuong c, int n);
void TinhTich_2_MaTran(MaTranVuong a, MaTranVuong b, MaTranVuong c, int n);
*/
//=====
//Định nghĩa các hàm
void NhapMaTran(MaTranVuong &a, int n, char kt)
{
    int i, j;
    for (i = 0; i < n; i++) // hàng i
        for (j = 0; j < n; j++) // cột j
        {
            cout << endl << kt << "[" << i << "]" << j << "] = ";
            cin >> *(a + i*n + j);
        }
}

void XuatMaTran(MaTranVuong a, int n)
{
    int i, j;
    for (i = 0; i < n; i++)
    {
        cout << endl << endl;
        for (j = 0; j < n; j++)
            cout << setw(4) << *(a + i*n + j);
    }
}

MaTranVuong TinhTong_2_MaTran(MaTranVuong a, MaTranVuong b, int n)
{
    int i, j;
    MaTranVuong c;
    c = new int[n*n];
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            *(c + i*n + j) = *(a + i*n + j) + *(b + i*n + j);
    return c;
}

MaTranVuong TinhHieu_2_MaTran(MaTranVuong a, MaTranVuong b, int n)
{
    MaTranVuong c;
    int i, j;
    c = new int[n*n];
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            *(c + i*n + j) = *(a + i*n + j) - *(b + i*n + j);

    return c;
}
```

```

MaTranVuong TinhTich_2_MaTran(MaTranVuong a, MaTranVuong b, int n)
{
    MaTranVuong c;
    int i, j, k;
    c = new int[n*n];
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
        {
            *(c + i*n + j) = 0;
            for (k = 0; k < n; k++)
                *(c + i*n + j) += *(a + i*n + k) * *(b + k*n + j);
        }
    return c;
}

```

//=====

### Bài 3. Xâu ký tự động

Viết chương trình tùy chọn thực hiện các chức năng sau đây trên chuỗi động :

0. Thoát khỏi chương trình
1. Nhập chuỗi
2. Xem chuỗi
3. Tính chiều dài chuỗi
4. Chèn ký tự x vào chuỗi tại vị trí cho trước
5. Xóa ký tự tại vị trí cho trước
6. Cắt ký tự cuối chèn vào vị trí đầu
7. Xóa tất cả các ký tự x

Yêu cầu:

- Cài đặt chuỗi động

Tạo project riêng, đặt tên **MaSV\_Lab09\_D\_Bai3**.

Chương trình tổ chức theo thư viện hàm và có hệ thống menu, phát triển như bài 1 mục D lab 8.

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

```

#define MAX 100
//Định nghĩa kiểu dữ liệu mới
typedef char *ChuoiDong;

//Khai báo nguyên mẫu

//Định nghĩa các hàm xử lý
//Tính chiều dài chuỗi
int TinhChieuDaiChuoi(ChuoiDong a)
{
    int i = 0;
    while (*(a+i) != NULL)
        i++;
    return i;
}

//Chen ky tu x vao chuoi a tai vi tri vt

```

```

//Input : a,x,vt
//output : 1; thành công; 0 : không thành công
int ChenKT_VT(ChuoiDong a, char x, int vt)
{
    int i, h, kq;
    h = TinhChieuDaiChuoi(a);
    if (vt < 0 || vt > h)
        kq = 0;
    else
    {
        for (i = h; i >= vt; i--)
            *(a + i + 1) = *(a + i);
        *(a + vt) = x;
        kq = 1;
    }
    return kq;
}

//Xoa ky tu tai vi tri vt cua chuoi a
//Input : a,vt
//output : 1; thành công; 0 : không thành công
int XoaKT_VT(ChuoiDong a, int vt)
{
    int i, h, kq;
    h = TinhChieuDaiChuoi(a);
    if (vt < 0 || vt > h - 1)
    {
        kq = 0;
    }
    else
    {
        for (i = vt; i < h; i++)
            *(a + i) = *(a + i + 1);
        kq = 1;
    }
    return kq;
}

//Cat ky tu cuoi chuoi roi chen vao tai vi tri dau chuoi
void CatCuoiChenDau(ChuoiDong a)
{
    int i, h;
    char x;
    h = TinhChieuDaiChuoi(a);
    x = *(a + h - 1);
    for (i = h - 2; i >= 0; i--)
        *(a + i + 1) = *(a + i);
    *(a + 0) = x;
}

//Xoa tat ca ky tu x trong chuoi
void Xoa_x(ChuoiDong a, char x)
{
    int i, h = 0;
    for (i = 0; a[i] != NULL; i++)

```



```

if (*(a + i) != x)
{
    *(a + h) = *(a + i);
    h++;
}
*(a + h) = NULL;
}

```

//=====

#### Bài 4. Cấu trúc động

Bảng điểm môn học của sinh viên chứa các thông tin sau :

- Mã sinh viên : chuỗi có đúng 7 ký tự
- Học và chữ lót của sinh viên : chuỗi có không quá 14 ký tự
- Tên sinh viên : chuỗi có không quá 7 ký tự
- Giới tính : chuỗi từ 2 đến 3 ký tự.
- Năm sinh : số nguyên dương 4 ký số
- Quê quán : chuỗi có không quá 14 ký tự
- Lớp : chuỗi có 5 ký tự
- Điểm : số thực từ 0 đến 10

Viết chương trình tùy chọn trên danh sách sinh viên với các chức năng :

0. Thoát khỏi chương trình
1. Tạo danh sách sinh viên
2. Xem danh sách sinh viên
3. Xuất danh sách sinh viên giảm dần theo điểm
4. Xem danh sách sinh viên theo lớp
5. Xuất danh sách sinh viên theo lớp và giảm dần theo điểm
6. Thống kê chất lượng học tập sinh viên theo lớp (Giỏi, Khá, TB, Yếu, Kém = ?)

Tiêu chuẩn xếp loại học tập của sinh viên (dựa vào điểm) như sau :

- **Giỏi** : Điểm  $\geq 8.5$
- **Khá** :  $7 \leq \text{Điểm} < 8.5$
- **TB** :  $5.5 \leq \text{Điểm} < 7$
- **Yếu** :  $4 \leq \text{Điểm} < 5.5$
- **Kém** : Điểm  $< 4$

**Yêu cầu:**

- Cài đặt mảng cấu trúc động
- Sử dụng bộ dữ liệu cho sẵn kèm theo sau đây:

Ma NU	Ho	Ten	GT	NS	Que quan	Lop	Diem
1512967	Trieu	Minh	Nu	1997	Ninh Thuan	CTK39	5.5
1410279	Hoang Duoc	Su	Nam	1995	Da Lat	CTK38	6.0
1512555	Au Duong	Phong	Nam	1996	Khanh Hoa	CTK39	7.5
1412120	Vo Thi	Yen	Nu	1996	Binh Dinh	CTK38	3.0
1313320	Le Ngoc	Minh	Nam	1995	Can Tho	CTK37	2.5
1510214	Dinh Thi	Yen	Nu	1997	Da Lat	CTK39	9.0
1512887	Vuong Ngoc	Yen	Nu	1997	Da Nang	CTK39	6.0
1414245	Vuong Trung	Duong	Nam	1996	Phu Yen	CTK38	7.0
1510192	Nhac Linh	San	Nu	1997	Da Lat	CTK39	3.2
1312890	Hoang	Dung	Nu	1995	Da Nang	CTK37	8.0
1510192	Cao Uien	Uien	Nu	1997	Da Lat	CTK39	9.6
1444405	Truong Uo	Ky	Nam	1996	Binh Dinh	CTK38	7.0
1412988	Ui Tieu	Bao	Nam	1996	Da Nang	CTK38	8.5
1312990	Duong	Qua	Nam	1995	Da Lat	CTK37	3.5
1333993	Chau Ba	Thong	Nam	1994	Quang Ngai	CTK37	9.1
1512128	Ta Van	Ton	Nam	1997	Binh Dinh	CTK39	9.3
1400128	Uien Thua	Chi	Nam	1997	Binh Thuan	CTK38	5.3
1512868	Doan	Du	Nam	1996	Da Lat	CTK39	8.5

**Bước 1.** Tạo 1 dự án Win32 Console Application mới. Đặt tên là **Lab09\_D\_Bai4**

**Bước 2.** Tạo cấu trúc cho chương trình như bài 1 mục D lab 8 (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 biểu thị số lượng tối đa sinh viên trong danh sách sinh viên, kiểu dữ liệu cấu trúc *SinhVien*

*//Định nghĩa hằng*

*//Định nghĩa hằng*

*#define MAX 100 //Số lượng tối đa sinh viên trong danh sách*

*#define NGANGDOI '=' //dấu bằng : gach 2 hàng*

*#define NGANGDON '-' //dấu trừ : gach 1 hàng*

*//Định nghĩa kiểu dữ liệu mới : kiểu cấu trúc Sinh viên*

*//Kiểu Sinh Viên*

*struct SinhVien*

*{*

*char maSV[8];*

*char hoLot[14];*

*char ten[7];*

*char gioiTinh[4];*

*unsigned int namSinh;*

*char queQuan[14];*

*char lop[7];*

*double diem;*

*};*

*//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. Tao danh sach sinh vien";*

*cout << "\n2. xem danh sach sinh vien";*

*cout << "\n3. Sap danh sach sinh vien giam dan theo diem";*

*cout << "\n4. xem danh sach sinh vien theo lop";*

*cout << "\n5. Xuat danh sach sinh vien theo lop va co diem giam dan";*

*cout << "\n6. Thong ke chat luong sinh vien theo lop";*

*cout << "\n=====";*

*}*

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. Ở đây ta dùng biến động mảng 1 chiều kiểu NhanVien và cài đặt bằng con trỏ .

```
void XuLyMenu(int menu, SinhVien *a, 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. Tao danh sach nhan vien";
            break;
        case 2:
            system("CLS");
            cout << "\n2. Xem danh sach nhan vien";
            break;
        case 3:
            system("CLS");
            cout << "\n3. Sap danh sach sinh vien giam dan theo diem";
            break;
        case 4:
            system("CLS");
            cout << "\n4. xem danh sach sinh vien theo lop";
            break;
        case 5:
            system("CLS");
            cout << "\n5. Xuat danh sach sinh vien theo lop va co diem giam dan";
            break;
        case 6:
```

```

        system("CLS");
        cout << "\n6. Thông kê chat lượng sinh viên theo lớp";
        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, SinhVien * a, 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 = 6,
        n = 0;
    SinhVien *a;
    a = new SinhVien[MAX];
    do
    {
        system("CLS");
        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 0 (thoát khỏi chương trình).

### **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 tạo và xuất danh sách sinh viên
- Trong tập tin **menu.h** bổ sung xử lý chức năng tạo, xuất dữ liệu trong hàm **XuLyMenu**.

- Trong tập tin **program.cpp** :

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

- Trong tập tin **thuvien.h** :

**Bổ sung các hàm tạo danh sách sinh viên và xuất danh sách sinh viên ra màn hình**

#### 4.1 Các hàm tạo dữ liệu

Ta dùng cách sau để tạo danh sách sinh viên theo bộ dữ liệu cho trước và tránh việc nhập từ bàn phím để khỏi mất thời gian.

```
//=====
//
//Tao Danh sach sinh vien
//=====
//Chen 1 sinh vien
void Chen_SV(char *maSV, char *hoLot, char *ten, char *gioiTinh, unsigned int namSinh,
             char *queQuan, char *lop, double diem, SinhVien *a, int &n)
{
    if (n < MAX)
    {
        strcpy_s((a + n)->maSV, 8, maSV);
        strcpy_s((a + n)->hoLot, 14, hoLot);
        strcpy_s((a + n)->ten, 7, 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);
        (a + n)->diem = diem;
        n++;
    }
}
```

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

```
//Tao Danh Sach sinh vien
void TaoDanhSachSinhVien(SinhVien *a, int &n)
{
    Chen_SV("1512967", "Trieu", "Minh", "Nu", 1997, "Ninh Thuan", "CTK39", 5.5, a, n);
    Chen_SV("1410279", "Hoang Duoc", "Su", "Nam", 1995, "Da Lat", "CTK38", 6, a, n);
    Chen_SV("1512555", "Au Duong", "Phong", "Nam", 1996, "Khanh Hoa", "CTK39", 7.5, a, n);
    Chen_SV("1412120", "Vo Thi", "Yen", "Nu", 1996, "Binh Dinh", "CTK38", 3, a, n);
    Chen_SV("1313320", "Le Ngoc", "Minh", "Nam", 1995, "Can Tho", "CTK37", 2.5, a, n);
    Chen_SV("1510214", "Dinh Thi", "Yen", "Nu", 1997, "Da Lat", "CTK39", 9, a, n);
    Chen_SV("1512887", "Vuong Ngoc", "Yen", "Nu", 1997, "Da Nang", "CTK39", 6, a, n);
    Chen_SV("1414245", "Vuong Trung", "Duong", "Nam", 1996, "Phu Yen", "CTK38", 7, a, n);
    Chen_SV("1510192", "Nhac Linh", "San", "Nu", 1997, "Da Lat", "CTK39", 3.2, a, n);
    Chen_SV("1312890", "Hoang", "Dung", "Nu", 1995, "Da Nang", "CTK37", 8, a, n);
    Chen_SV("1510192", "Cao Vien", "Vien", "Nu", 1997, "Da Lat", "CTK39", 9.6, a, n);
    Chen_SV("1444405", "Truong Vo", "Ky", "Nam", 1996, "Binh Dinh", "CTK38", 7, a, n);
    Chen_SV("1412988", "Vi Tieu", "Bao", "Nam", 1996, "Da Nang", "CTK38", 8.5, a, n);
    Chen_SV("1312990", "Duong", "Qua", "Nam", 1995, "Da Lat", "CTK37", 3.5, a, n);
    Chen_SV("1333993", "Chau Ba", "Thong", "Nam", 1994, "Quang Ngai", "CTK37", 9.1, a, n);
    Chen_SV("1512128", "Ta Van", "Ton", "Nam", 1997, "Binh Dinh", "CTK39", 9.3, a, n);
    Chen_SV("1400128", "Vien Thua", "Chi", "Nam", 1997, "Binh Thuan", "CTK38", 5.3, a, n);
    Chen_SV("1512868", "Doan", "Du", "Nam", 1996, "Da Lat", "CTK39", 8.5, a, n);
}
```

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

#### 4.2 Các hàm xuất danh sách sinh viên ra màn hình

```
//=====
//
```

////////////////////////////////// Xuất Danh sách sinh viên //////////////////////////////////

//=====

//Xuất ke ngang doi

void XuấtKeNgangDoi()

```
{
    int i;
    cout << "\n";
    cout << setiosflags(ios::left)
        << ':';
    for (i = 1; i <= 68; i++)
        cout << NGANGDOI;
    cout << ':';
}
```

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

//Xuất ke ngang don

void XuấtKeNgangDon()

```
{
    int i;
    cout << "\n";
    cout << setiosflags(ios::left)
        << ':';
    for (i = 1; i <= 68; i++)
        cout << NGANGDON;
    cout << ':';
}
```

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

//Xuất tiêu đề

void XuấtTieuDe()

```
{
    XuấtKeNgangDoi();
    cout << endl;
    cout << setiosflags(ios::left)
        << ':'
        << setw(7) << "Ma NV"
        << ':'
        << setw(15) << "Ho"
        << setw(7) << "Ten"
        << ':'
        << setw(4) << "GT"
        << ':'
        << setw(4) << "NS"
        << ':'
        << setw(14) << "Que quan"
        << ':'
        << setw(6) << "Lop"
        << ':'
        << setw(5) << "Diem"
        << ':';
}
```

```
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)
        << '\n'
        << setw(7) << p.maSV
        << '\n'
        << setw(15) << p.hoLot
        << setw(7) << p.ten
        << '\n'
        << setw(4) << p.gioiTinh
        << '\n'
        << setw(4) << p.namSinh
        << '\n'
        << setw(14) << p.queQuan
        << '\n'
        << setw(6) << p.lop
        << '\n'
        << setw(5) << setiosflags(ios::fixed) << setprecision(1) << p.diem
        << '\n';
}
```

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

//Xuat DS sinh vien

```
void Xuat_DSSV(SinhVien *a, 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.

4.3 Bổ sung nguyên mẫu các hàm :

```
void Chen_SV(char *maSV, char *hoLot, char *ten, char *gioiTinh, unsigned int namSinh,
             char *queQuan, char *lop, double diem, SinhVien *a, int &n);
void TaoDanhSachSinhVien(SinhVien *a, int &n);
```

```
void XuatKeNgangDoi();
void XuatKeNgangDon();
```



```
void XuatTieuDe();
void Xuat_1SV(SinhVien p);
void Xuat_DSSV(SinhVien *a, int n);
```

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

**- Trong tập tin *menu.h* :**

Bổ sung xử lý chức năng tạo danh sách sinh viên vào case 1, xem danh sách sinh viên vào case 2 (Các case từ 3 đến 6 giữ nguyên)

```
void XuLyMenu(int menu, SinhVien *a, int &n)
{
    switch (menu)
    {
        case 0:
            system("CLS");
            cout << "\n0. Thoat khỏi chương trình\n";
            delete []a;
            break;

        case 1:
            system("CLS");
            cout << "\n1. Tạo danh sách sinh viên";
            TaoDanhSachSinhVien(a, n);
            system("CLS");
            cout << "\n          DANH SACH SINH VIEN : \n";
            Xuat_DSSV(a, n);
            cout << "\nSố sinh viên trong danh sách : n = " << n;
            break;

        case 2:
            system("CLS");
            cout << "\n2. Xem danh sách sinh viên";
            cout << "\n          DANH SACH SINH VIEN : \n";
            Xuat_DSSV(a, n);
            cout << "\nSố sinh viên trong danh sách : 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.

Các bước tiếp theo, bổ sung định nghĩa các hàm chức năng trong tập tin *thuvien.h*, bổ sung xử lý chức năng trong hàm *XuLyMenu* trong tập tin *menu.h*

**Bước 5: Bổ sung chức năng 3 – sắp DSSV giảm dần theo điểm – vào chương trình**

- Trong tập tin *thuvien.h*, soạn thảo hàm sắp danh sách sinh viên giảm theo điểm
- Trong tập tin *menu.h* bổ sung xử lý chức năng sắp danh sách sinh viên giảm theo điểm trong hàm *XuLyMenu*.

**- Trong tập tin *thuvien.h* :**

**5.1 Hàm sắp danh sách sinh viên giảm theo điểm**

//Sap DSSV giam theo diem

```
void Sap_DSSV_GiamDiem(SinhVien *a, int n)
{
    SinhVien t;
    int i, j;
    if (n == 0)
    {
        cout << "\nDS rong!";
        return;
    }
    else
    {
        for (i = 0; i < n - 1; i++)
            for (j = i + 1; j < n; j++)
                if ((a + i)->diem < (a + j)->diem)
                {
                    t = *(a + i);
                    *(a + i) = *(a + j);
                    *(a + j) = t;
                }
    }
}
```

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

## 5.2 Bổ sung khai báo nguyên mẫu hàm

```
void Sap_DSSV_GiamDiem(SinhVien *a, int n);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng 3 vào case 3 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, SinhVien *a, int &n)
{
    switch (menu)
    {
        //...
        case 3:
            system("CLS");
            cout << "\n3. Sap danh sach sinh vien giam dan theo diem";
            cout << "\nDanh sach ban dau :\n";
            Xuat_DSSV(a, n);
            Sap_DSSV_GiamDiem(a, n);
            cout << "\nDanh sach sinh vien giam theo diem :\n";
            Xuat_DSSV(a, n);
            break;
        //...
    }
    _getch();
}
```

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

**Bước 6** *Bổ sung chức năng 4 – xem DSSV theo lớp– vào chương trình*

- Trong tập tin *thuvien.h* , soạn thảo các hàm xem danh sách sinh viên theo lớp
- Trong tập tin *menu.h* bổ sung xử lý chức năng sắp xếp xem danh sách sinh viên theo lớp trong hàm *XuLyMenu*.

**- Trong tập tin *thuvien.h* :**

**6.1 Các hàm xem danh sách sinh viên theo lớp**

```
//=====
//
/////////////// Danh sach lop ///////////////////
//=====
//

//Tao danh sach lop tu danh sach sinh vien : dsLop
void DSSV_Lop(SinhVien *a, int n, char lop[7], SinhVien *dsLop, int &h)
{
    int kq = n;
    int i;
    for (i = 0; i < n; i++)
        if (_strcmpi((a + i)->lop, lop) == 0)
        {
            kq = i;
            break;
        }
    if (kq == n)
        h = 0;
    else
    {
        h = 0;
        for (i = kq; i < n; i++)
            if (_strcmpi((a + i)->lop, lop) == 0)
            {
                *(dsLop + h) = *(a + i);
                h++;
            }
    }
}

//Xuat ds lop
void Xuat_DSSV_Lop(SinhVien *a, int n, char lop[7])
{
    SinhVien *dsLop;
    int i, h;
    dsLop = new SinhVien[MAX];
    DSSV_Lop(a, n, lop, dsLop, h);
    if (h == 0)
        cout << "\nKhong co lop " << lop << " trong DSSV";
    else
    {
        cout << "\n\nDanh sach sinh vien thuoc lop " << lop << " :\n";
        XuatTieuDe();
        for (i = 0; i < h; i++)
        {
            cout << endl;
        }
    }
}
```

```

        Xuat_1SV(*(dsLop + i));
    }
    XuatKeNgangDoi();
    cout << "\nCo " << h << " sinh vien thuoc lop " << lop;
}
delete[]dsLop;
}

```

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

## 6.2 Bổ sung khai báo nguyên mẫu hàm

```

void DSSV_Lop(SinhVien *a, int n, char lop[7], SinhVien *dsLop, int &h);
void Xuat_DSSV_Lop(SinhVien *a, int n, char lop[7]);

```

### - Trong tập tin *menu.h* :

Bổ sung xử lý chức năng 4 vào case 4 (Các case khác giữ nguyên)

```

void XuLyMenu(int menu, SinhVien *a, int &n)
{
    switch (menu)
    {
        //...
        case 4:
            system("CLS");
            cout << "\n4. Xuat danh sach sinh vien theo lop";
            cout << "\nDanh sach ban dau :\n";
            Xuat_DSSV(a, n);
            _getch();

            Xuat_DSSV_Lop(a, n, "CTK39");
            _getch();
            Xuat_DSSV_Lop(a, n, "CTK38");
            _getch();
            Xuat_DSSV_Lop(a, n, "CTK37");
            _getch();
            Xuat_DSSV_Lop(a, n, "CTK36");
            break;
        //...
    }
    _getch();
}

```

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

## **Bước 7.** *Bổ sung chức năng 5 – xuất DSSV theo lớp và có điểm giảm dần vào chương trình.*

- Trong tập tin *thuvien.h*, soạn thảo các hàm sắp danh sách sinh viên giảm dần theo điểm của các lớp
- Trong tập tin *menu.h* bổ sung xử lý chức năng xem danh sách sinh viên theo lớp và có điểm giảm trong hàm *XuLyMenu*.

### - Trong tập tin *thuvien.h* :

#### 7.1 Các hàm xuất danh sách sinh viên theo lớp và có điểm giảm dần

```
//=====//
//Xuất Danh sách lớp có điểm giảm dần ////////////////////////////////////////
//=====//
```

//Sắp xếp sinh viên thuộc lớp giảm dần theo điểm

```
void Sap_DSSV_Lop_GiamDiem(SinhVien *a, int n, char lop[7], SinhVien *dsLop, int &h)
{
    SinhVien t;
    int i, j;
    DSSV_Lop(a, n, lop, dsLop, h);
    if (h == 0)
        return; // Không có lớp trong DSSV
    else
    {
        for (i = 0; i < h - 1; i++)
            for (j = i + 1; j < h; j++)
                if ((dsLop + i)->diem < (dsLop + j)->diem)
                {
                    t = *(dsLop + i);
                    *(dsLop + i) = *(dsLop + j);
                    *(dsLop + j) = t;
                }
    }
}
```

//Xuất DS lớp giảm dần theo điểm

```
void Xuat_DSSV_Lop_Giam_Diem(SinhVien *a, int n, char lop[7])
{
    SinhVien *dsLop;
    int i, h;
    dsLop = new SinhVien[MAX];
    Sap_DSSV_Lop_GiamDiem(a, n, lop, dsLop, h);
    if (h == 0)
        cout << "\nKhông có lớp " << lop << " trong DSSV";
    else
    {
        cout << "\n\nDanh sách sinh viên thuộc lớp " << lop << " giảm dần theo điểm:\n";
        XuatTieuDe();
        for (i = 0; i < h; i++)
        {
            cout << endl;
            Xuat_1SV(*(dsLop + i));
        }
        XuatKeNgangDoi();
        cout << "\nCó " << h << " sinh viên thuộc lớp " << lop;
    }
    delete[] dsLop;
}
```

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

## 7.2 Bổ sung khai báo nguyên mẫu hàm

```
void Sap_DSSV_Lop_GiamDiem(SinhVien *a, int n, char lop[7], SinhVien *dsLop, int &h);
void Xuat_DSSV_Lop_Giam_Diem(SinhVien *a, int n, char lop[7]);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng 5 vào case 5 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, SinhVien *a, int &n)
{
    switch (menu)
    {
        //...
        case 5:
            system("CLS");
            cout << "\n5. Xuất danh sách sinh viên theo lớp và giảm dần theo điểm";
            cout << "\nDanh sách ban đầu :\n";
            Xuat_DSSV(a, n);
            Xuat_DSSV_Lop_Giam_Diem(a, n, "CTK39");
            _getch();
            Xuat_DSSV_Lop_Giam_Diem(a, n, "CTK38");
            _getch();
            Xuat_DSSV_Lop_Giam_Diem(a, n, "CTK37");
            _getch();
            Xuat_DSSV_Lop_Giam_Diem(a, n, "CTK36");
            //...
        }
    }
    _getch();
}
```

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

**Bước 8.** *Bổ sung chức năng 6 –thống kê chất lượng sinh viên theo lớp vào chương trình.*

- Trong tập tin *thuvien.h* , soạn thảo các hàm *thống kê chất lượng sinh viên theo lớp*
- Trong tập tin *menu.h* bổ sung xử lý chức năng *thống kê chất lượng sinh viên theo lớp* trong hàm *XuLyMenu*.

- Trong tập tin *thuvien.h* :

8.1 Hàm *thống kê chất lượng sinh viên theo lớp và xuất kết quả ra màn hình*

```
//=====//
//Thong ke theo lop chat luong sinh vien va xuat ket qua ra man hinh //
//=====//
```

```
void ThongKe_ChatLuong(SinhVien *a, int n, char lop[7])
{
    SinhVien *dsLop;
    int h = 0;
    dsLop = new SinhVien[MAX];
    DSSV_Lop(a, n, lop, dsLop, h);
    int i;

    int kem = 0,
        yeu = 0,
        trungBinh = 0,
        kha = 0,
        gioi = 0;
```

```
//Thong ke
if (h == 0)
{
    cout << "\nDanh sach " << lop << " rong!";
    _getch();
}
else
{
    for (i = 0; i < h; i++)
    {
        if ((dsLop + i)->diem >= 8.5)
            gioi++;
        else
            if (7 <= (dsLop + i)->diem && (dsLop + i)->diem < 8.5)
                kha++;
            else
                if (5.5 <= (dsLop + i)->diem && (dsLop + i)->diem < 7)
                    trungBinh++;
                else
                    if (4 <= (dsLop + i)->diem && (dsLop + i)->diem < 5.5)
                        yeu++;
                    else
                        if (0 <= (dsLop + i)->diem && (dsLop + i)->diem < 4)
                            kem++;
    }
    //Xuat ds lop tuong ung
    Xuat_DSSV_Lop(a, n, lop);
    //Xuat ket qua thong ke lop
    cout << "\nThong ke chat luong lop " << lop << " :\n";
    cout << "\nSo sinh vien gioi : Gioi = " << gioi << ", ti le (%): ti le = "
        << setiosflags(ios::fixed) << setprecision(1) << (double)gioi / h;
    cout << "\nSo sinh vien kha : Kha = " << kha << ", ti le (%): ti le = "
        << setiosflags(ios::fixed) << setprecision(1) << (double)kha / h;
    cout << "\nSo sinh vien TB : TB = " << trungBinh << ", ti le (%): ti le = "
        << setiosflags(ios::fixed) << setprecision(1) << (double)trungBinh / h;
    cout << "\nSo sinh vien yeu : Yeu = " << yeu << ", ti le (%): ti le = "
        << setiosflags(ios::fixed) << setprecision(1) << (double)yeu / h;
    cout << "\nSo sinh vien kem : kem = " << kem << ", ti le (%): ti le = "
        << setiosflags(ios::fixed) << setprecision(1) << (double)kem / h;
    }
}
```

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

## 8.2 Bổ sung khai báo nguyên mẫu hàm

```
void ThongKe_ChatLuong(SinhVien *a, int n, char lop[7]);
```

### - Trong tập tin *menu.h* :

Bổ sung xử lý chức năng 6 vào case 6 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, SinhVien *a, int &n)
{
    switch (menu)
```

```

{
    //...
    case 6:
        system("CLS");
        cout << "\n6. Thông kê chat luong lop";
        cout << "\nDanh sach ban dau :\n";
        Xuat_DSSV(a, n);
        _getch();
        system("CLS");
        ThôngKe_ChatLuong(a, n, "CTK39");
        _getch();
        ThôngKe_ChatLuong(a, n, "CTK38");
        _getch();
        ThôngKe_ChatLuong(a, n, "CTK37");
        _getch();
        ThôngKe_ChatLuong(a, n, "CTK36");
        break;
    //...
}
_getch();
}

```

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

Kiểm tra kết quả thực hiện chức năng 6.

**Kiểm tra tất cả chức năng chương trình – Kết thúc chương trình.**

## E. Bài tập bắt buộc

**Tất cả các bài tập phải được tổ chức dưới dạng thư viện hàm, có/không có hệ thống menu và cài đặt cấu trúc dữ liệu động.**

### Bài 1. Mảng động 1 chiều

Viết chương trình tùy chọn thực hiện các chức năng sau đây trên mảng số nguyên :

0. Thoát khỏi chương trình
1. Nhập tự động mảng
2. Xem mảng
3. Tính giá trị lớn nhất
4. Tính tổng các phần tử của mảng
5. Đếm các số dương trong dãy
6. Đếm số lần xuất hiện của x trong dãy

Yêu cầu:

- Cài đặt mảng động
- Các thuật toán cài đặt dưới hình thức đệ quy

### Bài 2. Xâu ký tự

Viết chương trình thực hiện các thao tác sau trên chuỗi động:

- Đếm số khoảng trắng (gồm cả ký tự cách và Tab) trong chuỗi.
- Tìm vị trí xuất hiện của chuỗi *t* trong chuỗi *s*. Nếu *s* không chứa *t* thì trả về -1.
- Tìm vị trí xuất hiện đầu tiên của ký tự *X* cho trước trong chuỗi *s*.



- Đảo vị trí của từ đầu và từ cuối trong chuỗi  $s$ . Ví dụ:  $s = \text{meo an ca}$  thì kết quả là  $s = \text{ca an meo}$
- Đổi ký tự đầu tiên trong chuỗi  $s$  sang chữ HOA, còn lại là chữ thường.
- Liệt kê từng ký tự và số lần xuất hiện của chúng trong chuỗi  $s$  (không phân biệt hoa-thường).
- Đếm số từ trong chuỗi  $s$ .

### Bài 3. Mảng 2 chiều động

Viết chương trình thực hiện các thao tác sau trên ma trận vuông động:

- Kiểm tra ma trận có phải là đối xứng
- Kiểm tra ma trận có phải là tam giác trên.
- Kiểm tra ma trận có phải là tam giác dưới.
- Kiểm tra ma trận có phải là tam giác chéo
- Kiểm tra ma trận có phải là tam giác đơn vị

### Bài 4. Quản lý sách

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

- Nhập danh sách tài liệu.
- Xem danh sách tài liệu
- 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.
- Liệt kê các tài liệu theo từng năm xuất bản.
- Xem thông tin tài liệu theo mã số tài liệu (**maTI**) 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.

## F. Bài tập làm thêm

### Bài 1. 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 (DSĐG) tăng dần theo tên, nếu trùng tên, sắp tăng theo họ.
3. Sắp xếp và xuất DSĐG 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 2. Bài cào

Bài cào là một kiểu chơi bài bằng bộ bài tây 52 lá. Bài được chia cho từng người, mỗi người 3 lá. Điểm của người chơi trong mỗi ván là số lẻ của tổng điểm 3 lá bài. Ví dụ, nếu tổng điểm 3 lá bài là 27 thì người đó được 7 điểm, nếu tổng là 10 điểm thì được 0 điểm. Cách tính điểm của các lá bài như sau:

- Các lá 2, 3, ..., 10 mỗi lá có điểm tương ứng với con số đó, bất kể lá bài màu gì.
- Lá A có điểm là 1, các lá J, Q, K đều được tính là 10 điểm.

Sau khi tính điểm và trình bài, ai có số điểm cao nhất là thắng ván đó. Trường hợp đặc biệt, ai sở hữu được cả 3 lá bài đều là bài tây (J, Q hoặc K) hoặc cả 3 lá đều cùng điểm số thì thắng ngay ván đó, không cần tính điểm. Nếu có từ 2 người trở lên có cùng điểm số cao nhất thì tiền cược được chia đều.

Viết chương trình minh họa trò chơi theo mô tả trên. Trong đó, máy tính đóng vai trò người chia bài. Sau mỗi ván, máy phải xáo bài trước khi chia. Có tất cả 10 người chơi, mỗi người được cấp một số tiền M. Chương trình sẽ dừng khi chỉ còn 1 người đủ tiền đặt cược hoặc khi người dùng chọn chức năng thoát chương trình. Tiền cược quy định cho mỗi ván là C với  $C \leq M/10$ . Những người chơi có số tiền bé hơn C không được phép tham gia tiếp.

Chương trình phải có các chức năng sau: thiết lập mức cược C, chia bài, tính điểm và thoát chương trình.

## Bài 3. Ma trận xoắn ốc

Viết chương trình cho phép người dùng nhập vào số nguyên  $n$ . Sau đó, xuất ra màn hình ma trận vuông cấp  $n$  sau khi đã điền các số từ 1 đến  $n^2$  theo chiều xoắn ốc như hình dưới đây.

- Tính trung bình cộng của các phần tử trong mảng
- Tính tổng bình phương của các phần tử trong mảng
- Tính độ lệch lớn nhất giữa 2 phần tử nằm liên tiếp nhau
- Tính tổng lớn nhất của  $k$  phần tử liên tiếp (có thể tính xoay vòng, ví dụ cho  $n=10, k=4$ , vị trí bắt đầu tính là 8 thì sẽ tính tổng của các phần tử thứ 8, thứ 9, thứ 0 và thứ 1).
- Tìm phần tử xuất hiện nhiều nhất và số lần xuất hiện của nó.

- Tìm số âm lớn nhất và vị trí của nó.
- Đếm và xuất các phần tử xuất hiện ít nhất  $k$  lần với  $k$  cho trước.
- Đếm số lần xuất hiện của phần tử  $x$  kể từ vị trí  $vt$  cho trước.
- Xáo trộn các phần tử trong mảng một cách ngẫu nhiên.
- Sắp các số nguyên tố nằm đầu mảng và tăng, các số còn lại nằm ở cuối và giảm dần.

TRƯỜNG