

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THỰC PHẨM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**



Giảng viên hướng dẫn: **Ngô Dương Hà**

Môn học**: Nhập môn lập trình**

Lớp: **12DHBM5**

Chủ đề: **18**

TPHCM, Tháng 2 Năm 2022

**I.Thực hiện xử lý trên mảng 2 chiều:**

*Câu 1: Tạo một ma trận A có kích thước m dòng x n cột (với m>0,n>0). Nhập/xuất A từ bàn phím, tạo ngẫu nhiên hoặc từ file test.*

\*Code:

#define \_CRT\_SECURE\_NO\_WARNINGS -> Tùy trình biên dịch mà sẽ có dòng này hoặc không,ở đây dùng Visual Studio 2022

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

int a[100][100], b[2][100];

int n, m;

Hàm nhập ma trận từ bàn phím

void Nhapmatran(int x[100][100], int& n, int& m)

{

do {

printf("\nNhap vao so luong dong: ");

scanf("%d", &n);

printf("\nNhap vao so luong cot: ");

scanf("%d", &m);

if (n < 0 || m < 0) {

printf("\nSo luong dong va cot phai lon hon 0. Nhap lai!");

}

} while (n < 0 || m < 0);

* Sử dụng vòng lặp do-while để nhập 2 giá trị n (số hàng), m(số cột) trong ma trận phải là giá trị dương, nếu số hàng (hoặc số cột) đã nhập là giá trị thì yêu cầu nhập lại.

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

printf("\nx[%d][%d]= ", i, j);

scanf("%d", &x[i][j]);

}

}

}

* 2 vòng for để quét từng vị trí số cột và số hàng nằm trong khoảng từ 0 đến m-1 hoặc n-1, sau đó để người dùng nhập vào giá trị tương ứng với từng vị trí đó

Hàm nhập ma trận với các giá trị được tạo ngẫu nhiên

void Nhapmatranrandom(int x[100][100], int &n , int &m){

    do{

        printf("Nhap vao so dong: ");

        scanf("%d", &n);

        printf("\nNhap vao so cot: ");

        scanf("%d", &m);

    } while (m <= 0 || n <= 0);

    srand(time(NULL));

    for (int i = 0; i < m; i++){

        for (int j = 0; j < n; j++){

            x[i][j] = rand()%100;

        }

    }

}

* Hàm rand() là hàm có sẵn trong thư viện stdlib.h, với chức năng tạo ra các giá trị ngẫu nhiên trong phạm vi từ 0 đến 2.147.483.647, tuy nhiên ở đây ta chỉ cho hàm tạo ngẫu nhiên các giá trị trong phạm vi từ 0 đến 100, và sau đó gán các giá trị này vào các vị trí tương ứng các hàng và cột trong ma trận.

Sau khi nhập xong ma trận thì ta sẽ có hàm xuất ma trận ra màn hình

void Xuatmatran(int x[100][100], int n, int m) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

printf("%d", x[i][j]);

printf("\t");

}

printf("\n");

}

}

* 2 vòng for trong hàm để là để quét từng vị trí tương ứng với số cột và số hàng sau đó sẽ in từng giá trị ra màn hình

*Câu 2: Tính giá trị trung bình cộng của các phần tử trên 4 biên, đường chéo chính và đường chéo phụ*

Hàm tính trung bình cộng các giá trị ở 4 biên của ma trận:

void Tbcbien(int x[100][100], int n, int m) {

int s = 0, dem = 0;

for (int j = 0; j < m; j++) { (1)

s += x[0][j];

dem++;

}

for (int i = 1; i < n; i++) { (2)

s += x[i][0];

dem++;

}

for (int j = 1; j < m; j++) { (3)

s += x[n - 1][j];

dem++;

}

for (int i = 1; i < n - 1; i++) { (4)

s += x[i][m - 1];

dem++;

}

float tb = (float)s / dem;

printf("\nTrung binh cong duong bien la: %.2f", tb);

}

* Ở hàm tính trung bình cộng, ta sẽ có 1 biến đếm (dem) và một biến tổng (s), đều khởi tạo giá trị đầu tiên là 0. Biến s (tức biến tổng), dùng để cộng tất cả các giá trị sau đó lưu vào chung 1 biến, biến đếm (dem) dùng để đếm số giá trị được cộng vào.
* Vòng lặp for đầu tiên (1), dùng để duyệt dòng đầu tiên, sau đó cộng từng giá trị ở dòng đầu tiên vào biến s và tăng biến dem.
* ở vòng lặp for thứ 2 (2), dùng để duyệt cột bên trái bắt đầu từ dòng thứ 2, sau đó cộng từng giá trị vào biến s và tăng biến dem.
* ở vòng lặp for thứ 3 (3), dùng để duyệt dòng cuối cùng bắt đầu từ cột thứ 2, sau đó cộng từng giá trị vào biến s và tăng biến dem.
* ở vòng lặp for thứ 4 (4),dùng để duyệt cột còn lại, sau đó cộng từng giá trị vào biến s và tăng biến dem.
* Cuối cùng, biến tb là biến để lấy giá trị trung bình cộng, vì giá trị của trung bình cộng buộc phải là số thực nên phải khai báo biên theo kiểu dữ liệu float, sau đó lấy kết quả của biến s chia lấy phần nguyên cho biến dem được thực hiện ở 4 vòng for trên, ta được giá trị trung bình cộng các giá trị ở 4 biên của ma trận và in ra màn hình

Hàm tính trung bình cộng các giá trị ở đường chéo chính của ma trận:

void Tbcheochinh(int x[100][100], int n, int m) {

int s = 0;

int dem = 0;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if (i == j) {

s += x[i][j];

dem++;

}

}

}

float tb = (float)s / dem;

printf("\nTrung binh cong duong cheo chinh la: %.2f", tb);

}

* Tương tự như hàm tính trung bình cộng các giá trị ở 4 biên của ma trận, ở hàm này, ta vẫn có 1 biến tổng (s) và 1 biến đếm(dem), 2 vòng lặp for bên trong hàm được sử dụng để quét toàn bộ ma trận, câu điều kiện bên trong vòng lặp for dùng để lọc ra các giá trị nằm trên đường chéo chính, sau đó cộng chúng lại với nhau và gán cho biến s, cứ mỗi lần như vậy sẽ cộng biến dem lên 1 đơn vị.
* Cuối cùng lấy kết quả của biến s và biến dem được thực hiện ở 2 vòng lặp for trên để tính trung bình cộng, sau đó gán cho biến tb.

Hàm tính trung bình cộng các giá trị ở đường chéo phụ của ma trận:

void Tbcheophu(int x[100][100], int n, int m) {

int s = 0;

int dem = 0;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if (i + j == m - 1) {

s += x[i][j];

dem++;

}

}

}

float tb = (float)s / dem;

printf("\nTrung binh cong duong cheo phu la: %.2f", tb);

}

* Tương tự như hàm tính trung bình cộng các giá trị ở 4 biên của ma trận, ở hàm này, ta vẫn có 1 biến tổng (s) và 1 biến đếm (dem), 2 vòng lặp for bên trong hàm được sử dụng để quét toàn bộ ma trận, câu điều kiện bên trong vòng lặp for dùng để lọc ra các giá trị nằm trên đường chéo phụ, sau đó cộng chúng lại với nhau và gán cho biến s, cứ mỗi lần như vậy sẽ cộng biến dem lên 1 đơn vị.
* Cuối cùng lấy kết quả của biến s và biến dem được thực hiện ở 2 vòng lặp for trên để tính trung bình cộng, sau đó gán cho biến tb.

*Câu 3: Liệt kê danh sách phần tử của từng cột chứa toàn bộ những số thịnh vượng.*

Để giải quyết yêu cầu bài toán, ta cần có 2 hàm, 1 hàm để kiểm tra những số thịnh vượng và hàm còn lại để liệt kê những số thịnh vượng trong ma trận

bool Checkthinhvuong(int a) {

int check = a;

int s = 0, c;

while (check != 0) {

c = check % 10;

s += c \* c \* c;

check /= 10;

}

if (s == a) {

return true;

}

else {

return false;

}

}

* Câu lệnh while đầu tiên dùng để tách từng chữ số riêng, sau đó lập phương lên và cộng lại với nhau,sau khi thực hiện xong sẽ xét đến câu lệnh điều kiện if nếu như tổng lập phương của từng chữ số bằng chính số đó, thì hàm sẽ trả về giá trị true, ngược lại sẽ trả về giá trị false

void Lietkethinhvuong(int x[100][100], int n, int m) {

for (int j = 0; j < m; j++) {

printf("\nCot thu %d:", j);

for (int i = 0; i < n; i++) {

if (Checkthinhvuong(x[i][j])) {

printf(" %d ", x[i][j]);

}

}

}

}

* Hàm này dùng để liệt kê các phần tử theo cột thỏa mãn điều kiện true của *hàm trên (Checkthinhvuong)*

*Câu 4: Hãy cho biết dòng có chứa giá trị trung bình nhân của các số nguyên tố lớn nhất.*

Để giải quyết yêu cầu, ta viết ra làm 3 hàm, tương ứng với 3 công đoạn giải quyết

Công đoạn đầu tiên ta sẽ viết hàm kiểm tra điều kiện là số nguyên tố, công đoạn thứ 2 là đối chiếu với điều kiện số nguyên tố để lọc ra các số nguyên tố sau đó lấy căn bậc n của n số nguyên tố đó ta được trung bình nhân của từng dòng, công đoạn thứ 3 dùng kĩ thuật đặt lính canh để tìm ra giá trị lớn nhất giữa các trung bình nhân đó

Code:

int KiemTraSoNguyenTo(int n) {

if (n < 2)

return 0;

for (int i = 2; i <= sqrt(n); i++) {

if (n % i == 0) {

return 0;

}

}

return 1;

}

double TrungBinhNhanNguyenToTungDong(int x[100][100], int n, int row) {

long tich = 1;

int dem = 0;

for (int i = 0; i < n; i++) {

if (KiemTraSoNguyenTo((a[row][i]))) {

tich \*= a[row][i];

dem++;

}

}

return pow(tich, 1 / (float)dem);

}

void MaxTrungBinhNhanNguyenTo(int x[100][100], int n, int m) {

double max = TrungBinhNhanNguyenToTungDong(x, m, 0);

int vitri;

for (int i = 0; i < n; i++) {

if (TrungBinhNhanNguyenToTungDong(x, m, i) > max) {

max = TrungBinhNhanNguyenToTungDong(x, m, i);

vitri = i;

}

}

printf("\nDong co trung binh nhan cac so nguyen to lon nhat la %d", vitri);

}

*Câu 5: Hãy hoán vị 2 cột c1 và c2 bất kì của ma trận A*

Để giải quyết yêu cầu bài toán ta chỉ cần 1 hàm để hoán đổi 2 cột với nhau

Code:

void Doigiatrihaicot(int x[100][100], int n, int m) {

int c1, c2;

printf("\nNhap vao hai cot can hoan doi: "); scanf("%d%d", &c1, &c2);

for (int i = 0; i < n; i++) {

int temp = x[i][c1];

x[i][c1] = x[i][c2];

x[i][c2] = temp;

Vòng lặp for này và những câu lệnh bên trong vòng lặp được dùng để hoán đổi từng phần tử ở cột c1 và cột c2 được nhập ở trên

}

}

*Câu 6: Từ ma trận A đã nhập, xuất ra một ma trận B sao cho từng dòng phải được sắp xếp tất cả các giá trị chẵn tăng dần và các giá trị lẻ giảm dần.*

Tạo biến chứa số lượng của tổng ma trận. Sau đó chạy for với điều kiện so sánh từng cặp số với nhau, nếu thỏa điều kiện 2 số cùng chẵn hoặc cùng lẻ và số sau bé hơn số trước thì hoán đổi vị trí, ngược lại giữ nguyên.

Code:

void Sapxepchantanglegiam(int a[100][100], int m, int n)

{

    int k = m \* n;

    for (int i = 0; i < k - 1; i++)

    {

        for (int j = i + 1; j < k; j++)

        {

            if(a[i / n][i % n] % 2 == 0 && a[j / n][j % n] % 2 == 0 && a[i/n][i%n]>a[j/n][j%n]

                || a[i / n][i % n] % 2 != 0 && a[j / n][j % n] % 2 != 0 && a[i/n][i%n]<a[j/n][j%n])

            {

                Swap(a[i / n][i % n], a[j / n][j % n]);

            }

        }

    }

}

*Câu 7: Từ ma trận A đã nhập, hãy tạo và xuất ra 2 ma trận C, D có cùng kích thước sao cho: C chứa toàn số dương và D chứa toàn số âm, các vị trí trống còn lại trên C,D sẽ để số 0*

Đầu tiên khởi tạo 2 ma trận C,D sao cho các phần tử ở các dòng và các cột đều mang giá trị 0, sau đó sẽ bắt đầu duyệt toàn bộ phần tử trong ma trận A đã nhập, nếu phần tử nào trong ma trận A mang giá trị dương, sẽ gán phần tử tại vị trí đó vào ma trận C. Ngược lại, phần tử nào trong ma trận A mang giá trị âm, sẽ gán phần tử tại vị trí đó vào ma trận D.

Code :

void Xuathaimatran(int x[100][100], int n, int m) {

int c[100][100]={0}, d[100][100]={0};

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if (x[i][j] > 0) {

c[i][j] = x[i][j];

}

else {

d[i][j] = x[i][j];

}

}

}

printf("\nMa tran duong: \n");

Xuatmatran(c, n, m);

printf("\nMa tran am: \n");

Xuatmatran(d, n, m);

}

*Câu 8: Tạo một ma trận E có kích thước là x dòng và y cột với điều kiện x,y>0 và x,y<m,n. Đếm số lần xuất hiện của ma trận E trong A.*

Code:

int AsubnetBmatrix(int A[][100], int B[][100], int aRow, int aCol, int bRow, int bCol)

{

int iARow, iACol;

int iBRow, iBCol;

bool flg;

int cnt = 0;

for (iBRow = 0; iBRow <= bRow - aRow; iBRow++)

{

for (iBCol = 0; iBCol <= bCol - aCol; iBCol++)

{

flg = true;

for (iARow = 0; iARow < aRow; iARow++)

{

for (iACol = 0; iACol < aCol; iACol++)

{

if (A[iARow][iACol] != B[iBRow + iARow][iBCol + iACol])

{

flg = false;

break;

}

}

if (flg == false)

{

break;

}

}

if (flg == true)

{

cnt++;

}

}

}

return cnt;

}

*Câu 9: Từ ma trận A đã nhập, hãy tạo và xuất ra ma trận F được dịch phải xoay vòng các cột theo trục đứng với chiều từ trái sang phải k lần.*

Để giải quyết yêu cầu, đầu tiên ta sẽ tạo ra mảng phụ (temp[100]), sau đó lưu lại cột bên trái ở ngoài cùng vào mảng phụ, rồi gán các cột còn lại từ trái qua phải, sau đó gán mảng phụ vào cột cuối bên trái (bên phải).

Code:

void Dichxoayvong(int x[100][100], int n, int m) {

int k;

printf("\nNhap vao so lan xoay: "); scanf("%d", &k);

int temp[100];

for (int e = 1; e <= k; e++) {

for (int i = 0; i < n; i++) {

temp[i] = x[i][m - 1];

}

for (int j = m - 1; j > 0; j--) {

for (int i = 0; i < n; i++) {

x[i][j] = x[i][j - 1];

}

}

for (int i = 0; i < n; i++) {

x[i][0] = temp[i];

}

}

Xuatmatran(x, n, m);

}

*Câu 10: Từ ma trận A đã nhập, tạo và xuất ra một ma trận I được xoắn ốc giảm dần theo ngược chiều kim đồng hồ.*

Đề giải quyết yêu cầu bài, đầu tiên ta cần viết hàm để chuyển mảng 2 chiều thành mảng 1 chiều, sau đó sẽ sắp xếp mảng 1 chiều đó theo chiều giảm dần, cuối cùng chuyển mảng 1 chiều thành mảng 2 chiều theo quy luật hình xoắn ốc ngược chiều kim đồng hồ.

Code:

void Giamdan(int x[100][100], int n, int m) {

int k = m \* n;

for (int i = 0; i < k - 1; i++) {

for (int j = i + 1; j < k; j++) {

if (x[i / m][i % m] < x[j / m][j % m]) {

int temp = x[i / m][i % m];

x[i / m][i % m] = x[j / m][j % m];

x[j / m][j % m] = temp;

}

}

}

}

void Chuyenmatranthanhmang1chieu(int x[100][100], int n, int m, int y[10000]) {

int index = 0;

int k = m \* n;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

y[index] = x[i][j];

index++;

}

}

for (int i = 0; i < k; i++) {

printf("%d ", y[i]);

}

}

void Xoanoc(int x[100][100], int n, int m) {

int z[100][100], p[1000];

int t = 0;

int h1 = 0, h2 = n - 1, c1 = 0, c2 = m - 1;

Giamdan(x, n, m);

Chuyenmatranthanhmang1chieu(x, n, m, p);

while (h1 <= h2 && c1 <= c2) {

for (int i = c2; i >= c1; i--) {

z[h1][i] = p[t];

printf("\n %d \n", z[h1][i]);

t++;

}

h1++;

for (int i = h1; i <= h2; i++) {

z[i][c1] = p[t];

t++;

}

c1++;

if (c1 <= c2 && h1 <= h2) {

for (int i = c1; i <= c2; i++) {

z[h2][i] = p[t];

t++;

}

h2--;

}

if (h1 <= h2 && c1 <= c2) {

for (int i = h2; i >= h1; i--) {

z[i][c2] = p[t];

t++;

}

c2--;

}

}

Xuatmatran(z, n, m);

}

**II.Mảng cấu trúc:**

*Câu 1: Xây dựng cấu trúc thông tin sinh viên như sau:*

* Mã số sinh viên là một chuỗi có tối đa 10 ký tự.
* Họ tên sinh viên là một chuỗi có tối đa 35 ký tự.
* Giới tính là một chuỗi có tối đa 5 ký tự.
* Hệ đào tạo chứa một trong các giá trị: ĐH chính quy, ĐH liên thông, Đh vừa làm vừa học.
* Khóa học là một chuỗi chứa tối đa 10 ký tự.Ví dụ: 11DHTH,11DHBM,12DHTH,12DHBM…
* Ngành học là một chuỗi có tối đa 30 ký tự
* Số tín chỉ tích lũy là một số nguyên
* Điểm trung bình tích lũy là một số thực.
* Danh sách học phần mà sinh viên đã học, mỗi học phần gồm các thông tin sau:

+ Mã học phần là một chuỗi tối đa 10 ký tự

+ Tên học phần là một chuỗi tối đa 50 ký tự

+ Số tín chỉ là một số nguyên từ 1 đến 12

+ Loại học phần: 0 - Lý thuyết,1 – Thực hành, 2: Thực tập – kiến tập, 3: Đồ án - Khóa luận.

+ Học kỳ là một số nguyên từ đến 12

+ Điểm tổng kết là một số thực từ 0 đến 10.

Code:

typedef struct sinhvien {

char ms[10];

char hoten[35];

char gioitinh[5];

int hedaotao;

char khoahoc[10];

char nganh[30];

int tongtinchi = 0;

float dtbtl;

int tonghocphan;

int hpdat = 0;

int hpkdat = 0;

int tongtctl = 0;

typedef struct hocphan {

char mhp[10];

char tenhp[50];

int sotc;

int loaihp;

int hk;

float diemtongket;

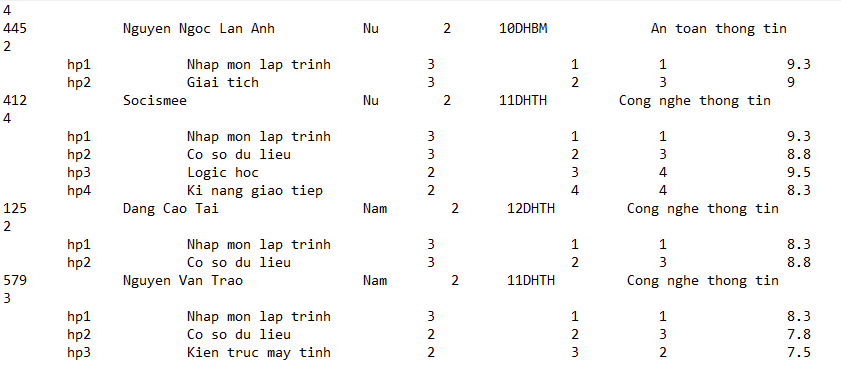
}HP;

HP hp[100];

}SV;

*Câu 2: Đọc thông tin sinh viên từ tập tin văn bản.*

Đầu tiên ta cần tạo 1 file txt để đọc



Code:

void Docfile(sinhvien a[100], int &slsv){

    FILE \*f;

    f=fopen("DSSV.txt", "rt");

    if(f==NULL){

        printf("Loi mo file!");

        exit(1);

    }

    fscanf(f, "%d", &slsv);

    for(int i=0; i<slsv; i++){

        fscanf(f, "\n%15s %30[^\n] %10s %7d %15s %35[^\n]",

        &a[i].ms, &a[i].hoten, &a[i].gioitinh, &a[i].hedaotao, &a[i].khoahoc, &a[i].nganh);

        fscanf(f, "\n%d", &a[i].tonghocphan);

        for(int j=0; j<a[i].tonghocphan; j++){

            fscanf(f, "\n%15s %30[^\n] %17d %10d %17d %10f",

            &a[i].hp[j].mhp, &a[i].hp[j].tenhp, &a[i].hp[j].sotc, &a[i].hp[j].loaihp, &a[i].hp[j].hk, &a[i].hp[j].diemtongket);

        }

    }

    fclose(f);

}

Sau đó ta đọc dữ liệu từ file theo các bước mở file đến đóng file.

*Câu 3: Tính số tín chỉ tích lũy và điểm trung bình tích lũy của sinh viên theo công thức đề cho:*

*Ta chạy for theo điều kiện điểm tổng kết của học phần lớn hơn hoặc bằng 4 thì tổng tín chỉ tích lũy cộng cho số tín chỉ của học phần*

*Chạy for tương tự với điểm tổng kết.*

Code:

void Tstc(sinhvien sv[100], int slsv){

    for (int i=0;i<slsv;i++){

        for (int j=0;j<sv[i].tonghocphan;j++){

            if (sv[i].hp[j].diemtongket>=4){

                sv[i].tongtctl=sv[i].tongtctl+sv[i].hp[j].sotc;

                sv[i].hpdat+=1;

            }

        }

        printf("Tong so tin chi tich luy cua sinh vien thu %d la %d\n", i+1, sv[i].tongtctl);

    }

    for (int i=0;i<slsv;i++){

        for (int j=0;j<sv[i].tonghocphan;j++){

            sv[i].dtbtl=sv[i].hp[j].diemtongket+sv[i].dtbtl;

        }

        sv[i].dtbtl=(float)(sv[i].dtbtl)/sv[i].tonghocphan;

        printf("Diem trung binh tich luy cua sv thu %d la %f\n",i+1,sv[i].dtbtl);

    }

}

*Câu 4: Xuất thông tin của các sinh viên ra màn hình theo dạng như sau:*

*STT MSSV Họ tên Giới tính Hệ ĐT STC tích lũy DTB Tích lũy*

*STT Mã HP Tên HP STC Loại HP Học kỳ ĐTK*

*Ta chạy for để xuất các thông tin ra. Ở hàm này chúng ta lưu ý để canh chỉnh hang để output được đều hơn, thêm dấu trừ phía trước để canh lề về bên trái.*

Code:

void Xuatthongtin(sinhvien a[100], int slsv){

    printf("\n--------------------------------------------------------------THONG TIN SINH VIEN------------------------------------------------------------------------------------\n");

    for(int i=0; i<slsv; i++){

        printf("\n%-10s %-15s %-30s %-10s %-10s %-15s %-30s %-17s %-17s",

        "STT", "MSSV", "Ho ten", "Gioi tinh", "He DT", "Khoa hoc", "Nganh",  "STC tich luy", "DTB tich luy");

        printf("\n%-10d %-15s %-30s %-10s %-10d %-15s %-30s %-17d %-17f",

        i+1, a[i].ms, a[i].hoten, a[i].gioitinh, a[i].hedaotao, a[i].khoahoc, a[i].nganh, a[i].tongtctl, a[i].dtbtl);

        printf("\n");

        printf("\n>>----- Danh Sach Hoc Phan: -----<< ");

        printf("\n");

        printf("\n\t%-10s %-15s %-30s %-17s %-10s %-17s %-10s",

        "STT", "Ma HP", "Ten HP", "So TC", "Loai HP", "Hoc ky", "DTK");

        for(int j=0; j<a[i].tonghocphan; j++){

            printf("\n\t%-10d %-15s %-30s %-17d %-10d %-17d %-10f",

            j+1, a[i].hp[j].mhp, a[i].hp[j].tenhp, a[i].hp[j].sotc, a[i].hp[j].loaihp, a[i].hp[j].hk, a[i].hp[j].diemtongket);

        }

        printf("\n-----------------------------------------------------------------------------------------------------------------------------------------------------------------\n");

    }

}

*Câu 5: Xuất thông tin kết quả học tập của một sinh viên khi biết mã số theo dạng như sau:*

*MSSV: Họ tên: Khóa học: Ngành học:*

*Số học phần đã học: Số học phần đạt: Số học phần không đạt:*

*Tổng số tín chỉ đã học: Điểm trung bình tích lũy:*

*Trong đó: học phần đạt là học phần có điểm tổng kết từ 4 trở lên, ngược lại là không đạt.*

*Ta khởi tạo biến để check với kiểu chuỗi giống với kiểu của mã số sinh viên. Sau đó so sanh chuỗi mới nhập với chuỗi mã số sinh viên, nếu bằng nhau thì xuất ra thông tin.*

Code:

void Checkmssv(sinhvien sv[100], int slsv){

    char check[10];

    printf("Nhap ma so can tim: ");

    getchar();

    fgets(check, sizeof(check), stdin);

    for (int i=0;i<slsv;i++){

        sv[i].hpkdat=sv[i].tonghocphan-sv[i].hpdat;

        if (strncmp(sv[i].ms, check, strlen(sv[i].ms)) == 0){

            printf("%-10s %-15s %-30s %-15s %-30s %-15s %-15s %-15s %-15s ","STT", "MSSV", "HoTen", "Khoa Hoc", "Nganh Hoc", "SHP dat", "SHP k dat", "Tong STC", "DTBTL");

            for(int j=0; j<sv[i].tonghocphan; j++){

                sv[i].tongtinchi=sv[i].hp[j].sotc+sv[i].tongtinchi;

            }

            printf("\n%-10d %-15s %-30s %-15s %-30s %-15d %-15d %-15d %-15f", i+1, sv[i].ms, sv[i].hoten, sv[i].khoahoc, sv[i].nganh, sv[i].hpdat, sv[i].hpkdat, sv[i].tongtinchi, sv[i].dtbtl);

        }

    }

}

*Câu 6: Xuất thông tin sinh viên có điểm trung bình tích lũy cao nhất và điểm trung bình tích lũy thấp nhất.*

*Chạy for để tìm điểm trung bình tích lũy lớn nhất và nhỏ nhất*

*Sau đó ta chạy for với điều kiện điểm trung bình tích lũy của sinh viên nào bằng với min và bằng với max thì xuất thông tin của sinh viên ấy ra*

Code:

void Tbtlmaxmin(sinhvien sv[100], int slsv){

    float max=0;

    float min=999;

    for (int i=0; i<slsv; i++){

        if(max<sv[i].dtbtl){

            max=sv[i].dtbtl;

        }

        if(min>sv[i].dtbtl){

            min=sv[i].dtbtl;

        }

    }

    for (int i=0; i<slsv; i++){

        if(sv[i].dtbtl==max){

            printf("Sinh vien co so diem cao nhat \n");

            printf("Ma so %s\n",sv[i].ms);

            printf("Ten %s\n",sv[i].hoten);

            printf("Khoa hoc %s\n",sv[i].khoahoc);

            printf("Nganh %s\n",sv[i].nganh);

        }

        if (sv[i].dtbtl==min){

            printf("Sinh vien co so diem thap nhat \n");

            printf("Ma so %s\n",sv[i].ms);

            printf("Ten %s\n",sv[i].hoten);

            printf("Khoa hoc %s\n",sv[i].khoahoc);

            printf("Nganh %s\n",sv[i].nganh);

        }

    }

}

*Câu 7: Thêm thông tin của một sinh viên vào danh sách.*

*Đầu tiên ta cần tạo hàm xóa xuống dòng để xóa xuống dòng ở kí tự kế cuối giúp thông tin nhập vào ko bị lỗi khi xuất file*

Code:

void Xoaxuongdong(char x[]){

    size\_t len = strlen(x);

    if(x[len-1]=='\n'){

        x[len-1]='\0';

    }

}

*Ta truyền tham chiếu để thay đổi giá trị của tổng số sinh viên (cụ thể là sẽ tăng lên 1 sau khi thêm 1 sinh viên vào)*

*Sau đó ta bắt đầu viết hàm nhập vào thông tin, lưu ý với dạng chuỗi sau mỗi lần nhập ta gọi hàm xóa xuống dòng để xóa kí tự xuống dòng. Ta chạy for để gán các giá trị vừa nhập vào tương ứng với thành phần trong struct*

Code:

void ThemSV(sinhvien sv[100], int &slsv){

    int pos=slsv;

    slsv=slsv+1;

    for(int i = pos; i < slsv; i++){

        printf("Nhap ms: ");

        fflush(stdin);

        fgets(sv[i].ms, sizeof(sv[i].ms), stdin);

        Xoaxuongdong(sv[i].ms);

        printf("\nNhap ho ten: ");

        fflush(stdin);

        fgets(sv[i].hoten, sizeof(sv[i].hoten), stdin);

        Xoaxuongdong(sv[i].hoten);

        printf("\nNhap gioi tinh: ");

        fflush(stdin);

        fgets(sv[i].gioitinh, sizeof(sv[i].gioitinh), stdin);

        Xoaxuongdong(sv[i].gioitinh);

        printf("\nNhap he dao tao: ");

        scanf("%d",&sv[i].hedaotao);

        printf("\nNhap khoa hoc: ");

        fflush(stdin);

        fgets(sv[i].khoahoc, sizeof(sv[i].khoahoc), stdin);

        Xoaxuongdong(sv[i].khoahoc);

        printf("\nNhap nganh hoc: ");

        fflush(stdin);

        fgets(sv[i].nganh, sizeof(sv[i].nganh), stdin);

        Xoaxuongdong(sv[i].nganh);

        printf("\nNhap vao so hoc phan cua sv thu %d da hoc:",i+1);

        scanf("%d",&sv[i].tonghocphan);

        for (int j=0;j<sv[i].tonghocphan;j++){

            printf("\nNhap ma hoc phan: ");

            fflush(stdin);

            fgets(sv[i].hp[j].mhp, sizeof(sv[i].hp[j].mhp), stdin);

            Xoaxuongdong(sv[i].hp[j].mhp);

            printf("\nNhap vao ten hoc phan: ");

            fflush(stdin);

            fgets(sv[i].hp[j].tenhp, sizeof(sv[i].hp[j].tenhp), stdin);

            Xoaxuongdong(sv[i].hp[j].tenhp);

            do{

                printf("\nNhap so tin chi ");

                scanf("%d",&sv[i].hp[j].sotc);

            }while (sv[i].hp[j].sotc>12 || sv[i].hp[j].sotc<1);

            printf("\nLoai hoc phan: ");

            scanf("%d",&sv[i].hp[j].loaihp);

            do{

                printf("\nNhap so hoc ki cua 1 hp: ");

                scanf("%d",&sv[i].hp[j].hk);

            }while (sv[i].hp[j].hk>12 || sv[i].hp[j].hk<1);

            do{

                printf("\nNhap diem tong ket cua hp: ");

                scanf("%f",&sv[i].hp[j].diemtongket);

            }while (sv[i].hp[j].diemtongket>10 || sv[i].hp[j].diemtongket<0);

        }

    }

}

*Câu 8: Thêm thông tin một học phần cho trước vào danh sách học phần của tất cả sinh viên.*

*Đầu tiên ta tạo các biến thay thế tương ứng với các giá trị liên quan đến học phần, sau đó ta nhập thông tin cần thêm.*

*Ta chạy for để gán các giá trị vừa tạo tương ứng vào các thành phần trong struct*

*Lưu ý với kiểu chuỗi ta không gán được nên dung hàm strcpy để gán chuỗi vừa nhập vào.*

Code:

void ThemHP(sinhvien sv[100], int slsv){

    char ma[10];

    char ten[35];

    int stc, loai, hk;

    float dtb;

    printf("\nMa hoc phan: ");

    fflush(stdin);

    fgets(ma, sizeof(ma), stdin);

    Xoaxuongdong(ma);

    printf("\nNhap ten hoc phan: ");

    fflush(stdin);

    fgets(ten, sizeof(ten), stdin);

    Xoaxuongdong(ten);

    printf("\nNhap so tin chi: ");

    do{

    scanf("%d",&stc);

    }while (stc>12);

    printf("\nNhap loai hoc phan: ");

    scanf("%d",&loai);

    printf("\nHoc ky: ");

    do{

        scanf("%d",&hk);

    }while (hk>12);

    printf("\nNhap diem tong ket cua mon: ");

    do{

    scanf("%f",&dtb);

    }while (dtb>10);

    for (int l=0; l<slsv; l++){

    int pos=sv[l].tonghocphan;

    sv[l].tonghocphan=sv[l].tonghocphan+1;

        for (int k=pos; k<sv[l].tonghocphan; k++){

            strcpy(sv[l].hp[k].mhp, ma);

            strcpy(sv[l].hp[k].tenhp, ten);

            sv[l].hp[k].sotc=stc;

            sv[l].hp[k].loaihp=loai;

            sv[l].hp[k].hk=hk;

            sv[l].hp[k].diemtongket=dtb;

        }

    }

}

*Câu 9: Xóa danh sách các học phần có điểm tích lũy nhỏ hơn 4 của sinh viên có mã số cho trước.*

*Tạo một biến để check thuôc cùng kiểu với mã số sinh viên rồi nhập mã số sinh viên cần tìm vào biến đó*

*Sau đó ta chạy for với điều kiện so sánh mã số sinh viên với biến để check, nếu bằng thì chạy tiếp for với điệu kiện điểm tổng kết của học phần đó phải nhỏ hơn 4 thì xóa học phần có điểm tổng kết nhỏ hơn 4 đi.*

Code:

void xoaHP(sinhvien sv[100], int slsv){

    char check[10];

    printf("\nNhap vao ma so:");

    fflush(stdin);

    fgets(check, sizeof(check), stdin);

    for (int i=0; i<slsv; i++){

        if (strncmp(sv[i].ms,check, strlen(sv[i].ms)) == 0){

            for (int j=0; j<sv[i].tonghocphan; j++){

                if(sv[i].hp[j].diemtongket<4){

                    for(int k = j; k < sv[i].tonghocphan - 1; k++){

                        strcpy(sv[i].hp[j].mhp, sv[i].hp[j+1].mhp);

                        strcpy(sv[i].hp[j].tenhp, sv[i].hp[j+1].tenhp);

                        sv[i].hp[j].sotc = sv[i].hp[j+1].sotc;

                        sv[i].hp[j].loaihp = sv[i].hp[j+1].loaihp;

                        sv[i].hp[j].hk = sv[i].hp[j+1].hk;

                        sv[i].hp[j].diemtongket = sv[i].hp[j+1].diemtongket;

                    }

                    --sv[i].tonghocphan;

                }

            }

        }

    }

}

*Câu 10: Xóa các sinh viên mà chưa học học phần nào (danh sách học phần rỗng).*

*Ta truyền tham chiếu để thay đổi giá trị của tổng số sinh viên (cụ thể là sẽ giảm đi sô sinh viên với danh sách học phần rỗng)*

*Ta chạy for với điều kiện nếu tổng học phần của sinh viên nào bằng 0 thì xóa sinh viên đó, vị trí của sinh viên bị xóa sẽ được thế bằng vị trí của sinh viên không bị xóa đứng sau nó.*

*Lưu ý với chuỗi ta dung hàm strcpy đến gán*

Code:

void xoaSVHProng(sinhvien sv[100], int &slsv){

    for (int i=0; i<slsv; i++){

        if(sv[i].tonghocphan==0){

            for(int k = i; k < slsv - 1; k++){

                strcpy(sv[i].ms, sv[i+1].ms);

                strcpy(sv[i].hoten, sv[i+1].hoten);

                strcpy(sv[i].gioitinh, sv[i+1].gioitinh);

                sv[i].hedaotao = sv[i+1].hedaotao;

                strcpy(sv[i].khoahoc, sv[i+1].khoahoc);

                strcpy(sv[i].nganh, sv[i+1].nganh);

                sv[i].tongtinchi = sv[i+1].tongtinchi;

                sv[i].dtbtl = sv[i+1].dtbtl;

                sv[i].tonghocphan = sv[i+1].tonghocphan;

                sv[i].hpdat = sv[i+1].hpdat;

                sv[i].hpkdat = sv[i+1].hpkdat;

                sv[i].tongtctl = sv[i+1].tongtctl;

                for (int j=0;j<sv[i].tonghocphan;j++){

                    strcpy(sv[i].hp[j].mhp, sv[i].hp[j+1].mhp);

                    strcpy(sv[i].hp[j].tenhp, sv[i].hp[j+1].tenhp);

                    sv[i].hp[j].sotc = sv[i+1].hp[j].sotc;

                    sv[i].hp[j].loaihp = sv[i+1].hp[j].loaihp;

                    sv[i].hp[j].hk = sv[i+1].hp[j].hk;

                    sv[i].hp[j].diemtongket = sv[i+1].hp[j].diemtongket;

                }

            }

            --slsv;

        }

    }

}

*Câu 11: Tính tổng số tín chỉ và tổng số học phần đã học của từng sinh viên.*

*Tạo biến tính tổng học phần và tổng tin chỉ*

*Chạy for để tính tổng học phần và tổng tín chỉ của từng sinh viên*

Code:

void Tongtctonghp(sinhvien sv[100], int tongsv){

    int tonghocphan=0;

    int tongtinchi=0;

    for(int i=0;i < tongsv;i++){

        for(int j=0;j<sv[i].tonghocphan;j++){

            sv[i].tongtinchi=sv[i].hp[j].sotc+sv[i].tongtinchi;

        }

        tonghocphan=sv[i].tonghocphan+tonghocphan;

        tongtinchi=sv[i].tongtinchi+tongtinchi;

        printf("\nTong tin chi cua sinh vien thu %d la: %d", i+1, tongtinchi);

        printf("\nTong hoc phan da hoc cua sinh vien thu %d la: %d", i+1, tonghocphan);

    }

}

*Câu 12: Đếm số sinh viên đã đậu học phần có tên là “Nhập môn lập trình” (điểm tổng kết > 4) và xuất ra thông tin của các sinh viên này.*

*Tạo biến đếm sau đó chạy for với điều kiện so sánh 2 chuỗi, nếu thỏa điều kiện biến đếm tăng lên 1.*

*Cho điều kiện nếu biến đếm lớn hơn 1 thì chạy for với điệu kiện tương tự như trên để xuất thông tin sinh viên*

Code:

void Demsohs(sinhvien sv[100], int tongsv){

    int check=0;

    for (int i=0;i<tongsv;i++){

        for (int j=0;j<sv[i].tonghocphan;j++){

            if ((strncmp(sv[i].hp[j].tenhp,"nhap mon lap trinh", strlen("nhap mon lap trinh")) == 0 || strncmp(sv[i].hp[j].tenhp,"Nhap mon lap trinh", strlen("Nhap mon lap trinh")) == 0) && sv[i].hp[j].diemtongket>4){

                check++;

            }

        }

    }

    printf("Co tong cong %d sv dau\n",check);

    if(check>0){

        printf("\n%-10s %-15s %-30s %-10s %-10s %-17s %-17s",

        "STT", "MSSV", "Ho ten", "Gioi tinh", "He DT", "STC tich luy", "DTB tich luy");

        for (int i=0;i<tongsv;i++){

            for (int j=0;j<sv[i].tonghocphan;j++){

                if ((strncmp(sv[i].hp[j].tenhp,"nhap mon lap trinh", strlen("nhap mon lap trinh")) == 0 || strncmp(sv[i].hp[j].tenhp,"Nhap mon lap trinh", strlen("Nhap mon lap trinh")) == 0) && sv[i].hp[j].diemtongket>4){

                    printf("\n%-10d %-15s %-30s %-10s %-10d %-17d %-17f",

                    i+1, sv[i].ms, sv[i].hoten, sv[i].gioitinh, sv[i].hedaotao, sv[i].tongtctl, sv[i].dtbtl);

                }

            }

        }

    }

}

***Câu 13: Cho biết điểm tổng kết cao nhất của môn “Nhập môn lập trình” và thông tin sinh viên có điểm cao nhất.***

*Tạo biến max với giá trị 0 để thế với các điểm tổng kết cao hơn.*

*Chạy for với 2 điều kiện là so sánh hai chuỗi và max nhỏ hơn điểm tổng kết thì gán giá trị của điểm tổng kết vào max.*

*Xuất ra giá trị điểm tổng kết*

*Chạy for với điều kiện so sánh 2 chuỗi như trên và điều kiện nếu điểm tổng kết của sinh viên nào bằng với max thì xuất thông tin sinh viên đó ra*

Code:

void Maxnmlt(sinhvien sv[100], int tongsv){

    float max=0;

    for (int i=0;i < tongsv;i++){

        for (int j=0;j<sv[i].tonghocphan;j++){

            if (strncmp(sv[i].hp[j].tenhp,"nhap mon lap trinh", strlen("nhap mon lap trinh")) == 0 || strncmp(sv[i].hp[j].tenhp,"Nhap mon lap trinh", strlen("Nhap mon lap trinh")) ==  0){

                if(max<sv[i].hp[j].diemtongket){

                    max = sv[i].hp[j].diemtongket;

                }

            }

        }

    }

    printf("\nDiem cao nhat la: %.2f", max);

    printf("\n%-10s %-15s %-30s %-10s %-10s %-17s %-17s",

    "STT", "MSSV", "Ho ten", "Gioi tinh", "He DT", "STC tich luy", "DTB tich luy");

    for (int i=0;i < tongsv;i++){

        for (int j=0;j<sv[i].tonghocphan;j++){

            if (strncmp(sv[i].hp[j].tenhp,"nhap mon lap trinh", strlen("nhap mon lap trinh")) == 0 || strncmp(sv[i].hp[j].tenhp,"Nhap mon lap trinh", strlen("Nhap mon lap trinh")) ==  0){

                if (sv[i].hp[j].diemtongket==max){

                    printf("\n%-10d %-15s %-30s %-10s %-10d %-17d %-17f",

                    i+1, sv[i].ms, sv[i].hoten, sv[i].gioitinh, sv[i].hedaotao, sv[i].tongtctl, sv[i].dtbtl);

                }

            }

        }

    }

}

*Câu 14: Sắp xếp danh sách sinh viên tăng dần theo tên, nếu cùng tên thì sắp giảm dần theo điểm trung bình tích lũy. Sau đó xuất kết quả ra*

*màn hình gồm: STT, MSSV, Họ tên, Giới tính, Hệ ĐT, STC tích lũy, DTB Tích lũy.*

*Tạo 2 biến cục bộ chạy for*

*Chạy for với điều kiện so sánh họ tên của sinh viên trước và sau, nếu lớn hơn 0 thì tạo biến kiểu struct sinh viên rồi đổi chỗ vị trí 2 sinh viên đó*

*Nếu điệu kiện không thỏa thì tiếp điều kiện 2 là hai chuỗi bằng nhau thì điệu kiện so sánh là điểm trung bình tích lũy*

*Và đổi chỗ tương tự như trên với biến kiểu struct*

Code:

void sapXep(SV sv[100], int slsv){

    int i, j;

    for(i = 0; i < slsv - 1; ++i){

        for(j = i + 1; j < slsv; ++j){

            if(strcmp(sv[i].hoten, sv[j].hoten) > 0){

                SV s = sv[i];

                sv[i] = sv[j];

                sv[j] = s;

            }

            else if(strcmp(sv[i].hoten, sv[j].hoten) == 0){

                if(sv[i].dtbtl < sv[j].dtbtl){

                    SV s = sv[i];

                    sv[i] = sv[j];

                    sv[j] = s;

                }

            }

        }

    }

}

*Câu 15: Cập nhật số tín chỉ học phần “Cơ sở dữ liệu” của các sinh viên khóa “11DHTH” từ 2 tín chỉ thành 3 tín chỉ.*

*Tạo biến chạy cục bộ*

*Chạy for với điều kiện so sánh hai chuỗi khóa học bằng nhau thì chạy tiếp for với điều kiện so sánh tên học phần*

*Nếu tồn tại giá trị phụ hợp thì gán 3 vào số tín chỉ ở học phần đó*

Code:

void capNhat(SV sv[100], int slsv){

    int i, j;

    for(i = 0; i < slsv; ++i){

        if(strncmp(sv[i].khoahoc, "11DHTH", strlen("11DHTH")) == 0){

            for(j = 0; j < sv[i].tonghocphan; ++j){

                if(strncmp(sv[i].hp[j].tenhp, "co so du lieu", strlen("co so du lieu")) == 0 || strncmp(sv[i].hp[j].tenhp, "Co so du lieu", strlen("Co so du lieu")) == 0){

                    sv[i].hp[j].sotc = 3;

                }

            }

        }

    }

}

*Ngoài ra ta còn cần tạo 1 hàm menu để đưa ra các chọn lựa ứng với số phụ hợp*

Code:

void Menu(){

    printf("\n===================MENU===================\n");

    printf("\n1/ Nhap xuat ma tran A, chon nhap tu phim/random/filetext ");

    printf("\n2/ Tinh trung binh bien, cheo chinh, cheo phu cua ma tran A");

    printf("\n3/ Liet ke danh sach cua tung cot chua so thinh vuong cua ma tran A");

    printf("\n4/ Tim dong co trung binh nhan cac so nguyen to lon nhat cua ma tran A");

    printf("\n5/ Hoan vi hai cot bat ki cu ma tran A");

    printf("\n6/ Tu A xuat ra B thanh hai dong sao cho moi dong la chan tang dan le giam dan");

    printf("\n7/ Tao hai ma tran moi tu A voi C duong D am");

    printf("\n8/ Tao ma tran E co x dong y cot, dem so lan xuat hien cua E trong A");

    printf("\n9/ Tu A tao ma tran F dich phai xoay vong tu trai sang phai");

    printf("\n10/ Tu A tao ma tran I xoan oc giam dan nguoc chieu kim dong ho");

    printf("\n11/ Tinh so tin chi tich luy va diem trung binh tich luy cua sinh vien");

    printf("\n12/ Xuat thong tin cua cac sinh vien ra man hinh");

    printf("\n13/ Xuat thong tin ket qua hoc tap cua sinh vien khi biet ma so");

    printf("\n14/ Xuat thong tin sinh vien co diem trung binh tich luy cao nhat va thap nhat");

    printf("\n15/ Them thong tin cua 1 sinh vien vao danh sach");

    printf("\n16/ Them thong tin hoc phan da cho truoc vao danh sach hoc phan cua tat ca sinh vien");

    printf("\n17/ Xoa danh sach cac hoc phan co diem tich luy nho hon 4 cua sinh vien co ma so cho truoc");

    printf("\n18/ Xoa cac sinh vien chua co hoc phan nao");

    printf("\n19/ Tinh tong so tin chi va tong so hoc phan da hoc cua sinh vien");

    printf("\n20/ Dem so sinh vien da dau hoc phan co ten 'Nhap mon lap trinh' va xuat thong tin cua cac sinh vien nay");

    printf("\n21/ Cho biet diem tong ket cao nhat cua hoc phan 'Nhap mon lap trinh' va thong tin sinh vien co diem cao nhat");

    printf("\n22/ Sap xep danh sach sinh vien tang dan theo ten, cung ten thi xep theo trung binh tich luy giam dan roi xuat ra man hinh");

    printf("\n23/ Cap nhat so tinh chi 'Co so du lieu' cua sinh vien khoa '11DHTH' tu 2 thanh 3 tin chi");

    printf("\n=========================================");

}

*Trong hàm main ta cần có biến để nhập lựa chọn và hàm do while trong đó chưa switch case để chọn lựa các yêu cầu muốn thức hiện*

Code:

int main(){

    int chon;

    struct sinhvien sv[100];

    int slsv;

    Docfile(sv,slsv);

    do{

        Menu();

        printf("\nChon bai (chon 0 de thoat): "); scanf("%d", &chon);

        switch(chon){

            case 1:

                int bai;

                printf("\n1/ Nhap tu ban phim");

                printf("\n2/ Nhap random");

                printf("\n3/ Nhap tu file text");

                printf("\nChon yeu cau: "); scanf("%d", &bai);

                switch(bai){

                    case 1:

                        Nhapmatran(a,n,m);

                        Xuatmatran(a,n,m);

                        break;

                    case 2:

                        Nhapmatranrandom(a,n,m);

                        Xuatmatran(a,n,m);

                        break;

                    case 3:

                        Nhapmangtufile(a,n,m);

                        break;

                }

                break;

            case 2:

                Tbcbien(a,n,m);

                Tbcheochinh(a,n,m);

                Tbcheophu(a,n,m);

                break;

            case 3:

                Lietkethinhvuong(a,n,m);

                break;

            case 4:

                MaxTrungBinhNhanNguyenTo(a,n,m);

                break;

            case 5:

                Doigiatrihaicot(a,n);

                Xuatmatran(a,n,m);

                break;

            case 6:

                Sapxepchantanglegiam(a,n,m);

                Xuatmatran(a,n,m);

                break;

            case 7:

                Xuathaimatran(a,n,m);

                break;

            case 8:

                fflush(stdin);

                printf("\nNhap vao so dong cua ma tran E: ");

                scanf("%d", &Arow);

                printf("\nNhap vao so cot cua ma tran E: ");

                scanf("%d", &Acol);

                ImportData(E, Arow, Acol);

                PrintMatrix(E, Arow, Acol);

                res = AsubnetBmatrix(E, a, Arow, Acol,n,m);

                if (res)

                printf("\nMa tran con da nhap xuat hien %d lan trong ma tran da nhap", res);

                else

                    printf("\nMa tran con da nhap khong phai la ma tran con");

                break;

            case 9:

                Dichxoayvong(a,n,m);

                break;

            case 10:

                Xoanoc(a,n,m);

                break;

            case 11:

                Tstc(sv, slsv);

                break;

            case 12:

                Xuatthongtin(sv, slsv);

                break;

            case 13:

                Checkmssv(sv, slsv);

                break;

            case 14:

                Tbtlmaxmin(sv, slsv);

                break;

            case 15:

                ThemSV(sv, slsv);

                break;

            case 16:

                ThemHP(sv,slsv);

                break;

            case 17:

                xoaHP(sv, slsv);

                break;

            case 18:

                xoaSVHProng(sv, slsv);

                break;

            case 19:

                Tongtctonghp(sv, slsv);

                break;

            case 20:

                Demsohs(sv, slsv);

                break;

            case 21:

                Maxnmlt(sv, slsv);

                break;

            case 22:

                sapXep(sv, slsv);

                break;

            case 23:

                capNhat(sv, slsv);

                break;

        }

    }while (chon!=0);

}