 ĐỀ SỐ 41

Câu 1. \* Trình bày thuật toán (bao gồm: input - output - process – bộ dữ liệu test) để tính trung bình cộng các số lẻ của một dãy n số gồm: x1, x2, ..., xn (0<=n<=50)

Câu 2. \* Viết chương trình nhập vào số có 4 chữ số. In ra màn hình chữ số hàng đơn vị của sốvừa được tử bàn phím

Câu 3, \*\* Viết chương trình nhập R từ bàn phím, Thực hiện tính diện tích và thể tích của hình cầu với bán kính R theo công thức: S =4\*R\*R;V=R\*S/3.

Câu 4. \* Hãy viết chương trình nhập n từ bàn phím và tính n!!. Sau đó in kết quả ra màn hình.

Câu 5. \*\* Viết hàm thống kê số các chữ số 0, số chữ số 1,..., số chữ số 9 trong chuỗi ký tự. Sau

đó, viết chương trình nhập một chuỗi bất kì, gọi hàm trên và cho biết kết quả. Không sử dụng thư viện có sẵn của C/C+

Câu 6. \* Viết hàm tách từ cuối cùng của xâu. Sau đó, viết chương trình nhập chuỗi họ và tên của bạn từ bàn phím và hiện tên của bạn lên màn hình. Không sử dụng thư viện có sẵn của C/C++

Câu 7. \*\* Viết các hàm thực hiện

• Nhập mảng gần n số nguyên (0<=n<=50)

• Hiện mảng gồm n số nguyên ra màn hình

• Xóa một phần tử tại vị trí k của mảng số nguyên

• Xóa các phần tử chia hết cho 3 trong mảng số nguyên

• Đếm số lượng các phần tử khác nhau trong mảng số nguyên.

-Viết chương trình chính Áp dụng lần lượt các hàm trên

Câu 8. \* Viết các hàm thực hiện công việc sau:

• Nhập ma trận số nguyên kích thước n\*m

• Hiện ma trận vừa nhập ra màn hình

• Hiện ra hàng có số lượng phân tử âm như âm nhất.

•Kiểm tra một số chính phương

• Liệt kê vị trí (dạng (i,j)) của các phần tử là số chính phương trong ma trận

• Đếm số lượng phần tử có giá trị bằng phần tử tại vị trí (0,0) của ma trận.

-Viết chương trình chính Áp dụng Tâm lượt các hàm trên.

Câu 9. " viết các chương trình con thực hiện các công việc sau:

o nhập danh sách các sách gồm các thông tin: tên sách, tên tác giả, nhà xuất bản, năm xuất bản;

o hiện danh sách vừa nhập ra màn hình; o sắp xếp danh sách các sách theo thứ tự alphabet của tên sách và ghi vào tệp sach txt; o thống kê các sách của tác giả x in ở nhà xuất bản y với x, y nhập từ bàn phím.

Viết chương trình chính để gọi các chương trình con theo trật tự trên. Nếu có thể tổ chức chươngtrình thông qua menu thì sẽ là một điểm cộng.

Câu 10. \* viết các chương trình con thực hiện các công việc sau: o nhập danh sách cán bộ gồm tên, phòng ban, hệ số lương, lương, trong đó lương= hệ số lương lương cơ bản, lương cơ bản được nhập vào từ bàn phím. o hiển thị thông tin các cán bộ vừa nhập ra màn hình. o hiện danh sách cán bộ có hệ số lương >5.0 ra màn hình và danh sách này vào tệp canbo .dat (dạngtập nhị phân)

o sắp xếp danh sách danh sách cán bộ theo thứ tự tăng dần của lương.

o đếm số cán bộ có lương trên 2 triệu và hiển thị ra màn hình (hoặc trả vềđiều kiện)số lượng cán bộ thoả mãn

Viết chương trình chính để gọi các chương trình con trên theo trật tự được viết như trên. Nếu có thể tổ chức chương trình thông qua menu thì sẽ là một điểm công.

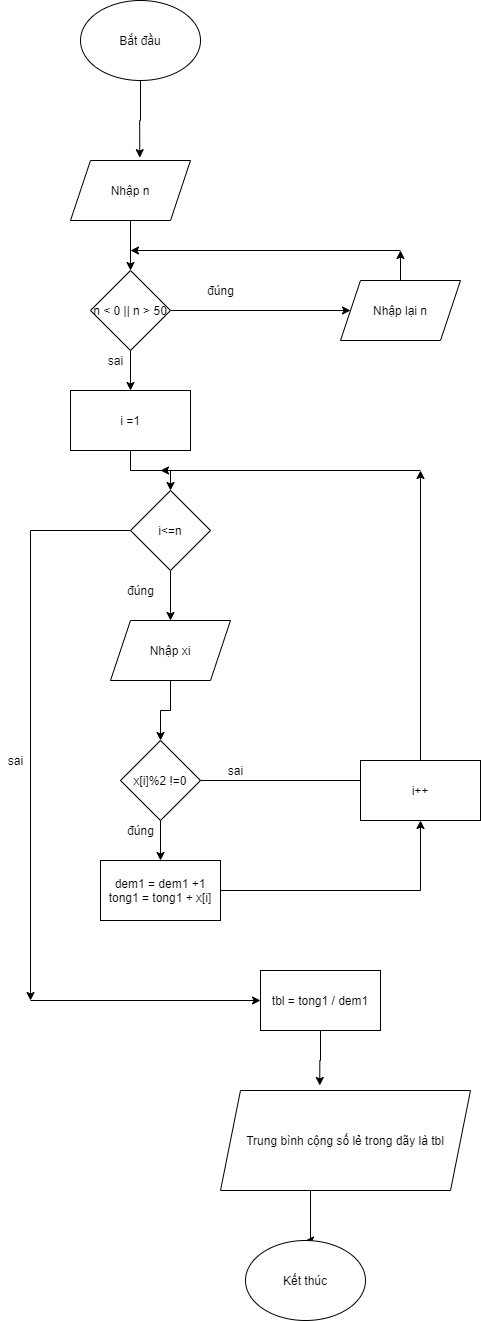
Bài Tập lớn

**Bài 1:**

a:)input: dãy n số

b:)output:trung bình cộng số lẻ

c:)process:



d:) bộ dữ liệu tets

\_nhập n=60-> sai: nhập lại n=5 -> đúng: nhập x1=1, x2=2, x3=3,x4=4, x5=5 -> trung bình lẻ: tbl=3

\_nhập n=3-> thỏa mãn: nhập x1=5, x2=6, x3=7 -> trả về tbl=6

**Bài 2:**

a:)input:nhập n 4 chữ số

b:)output:chữ số hàng đơn vị

c:)Process:

+Bắt đầu :

+B1:khai bao biến cần dùng ( n nguyên) nhập giá trị <n>

+B2:Ta dùng while để xét điều kiện <n> nếu < n > còn nhỏ hơn 1000 hoặc lơn hơn 9999 thì chương

trình còn "nhập lại n"

+B3 :khi <n> không thỏa mãn điều kiện B2 , thoát câu lệnh và thực hiện cho giá trị i=(n%1000) %10 để

lấy chứ số hàng đơn vị (ở đây tả thực hiện toán tử chia lấy phần dư)

+B4chương trình hiện giá trị : i (hay output của đề bài)

+chương trình kết thúc

d:)code:

#include<iostream>

using namespace std;

int main( ){

int n,i,j;

cout<<"nhap n 4 chu so:";

cin>>n;

while(n<1000 | n>9999){

cout<<"nhap lai n 4 chu so:";

cin>>n;

}

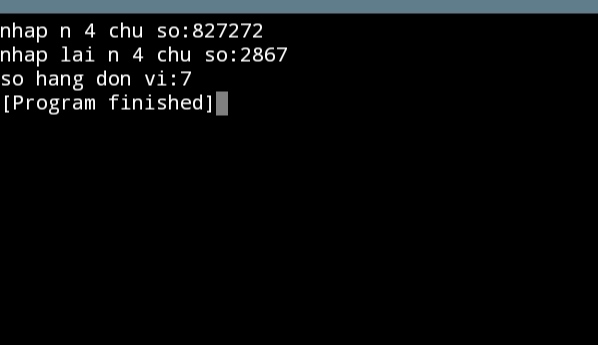
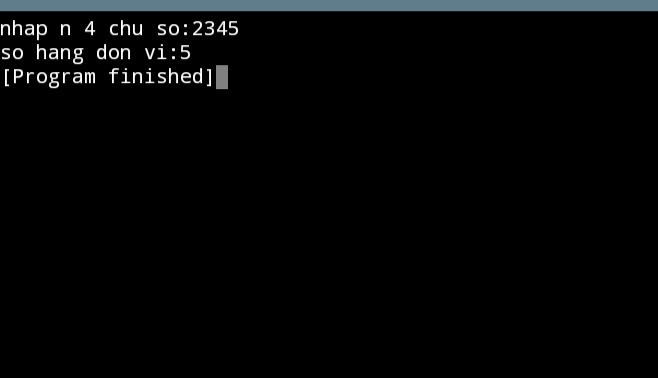
i=(n%1000)%10;

cout<<"so hang don vi:"<<i;

return 0;

}

e:) Test



**Bài 3:**

a:) input : nhập r

b:) output:giá trị S, V

c:) process:

+B1:bắt đầu khai bao biến ( r, S, V kiểu float) nhập giá trị r

+B2:thực hiện toán tử tính S=4\*r\*r và V=r\*S/3

+B3:hiện ra màn hình giá trị S, V

+kết thúc chương trình

d:) code:

#include<iostream>

#include<iomanip>

using namespace std;

int main(){

float S,V,r;

cout<<"nhap r:";

cin>>r;

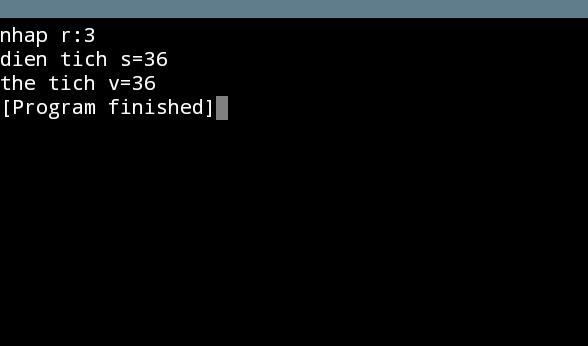
S=4\*r\*r;

V=r\*S/3;

cout<<"dien tich s="<<S<<endl;

cout<<"the tich V="<<V; }

e:) Test



**Bài 4:**

a:) input:nhập n

b:) output: n!!

c:) process:

-Bắt đầu

-B1:<n> nguyên

-B2: n%2==0

-2.1:Nếu đúng:

+2.1.1:cho giá trị i=2,

+2.1.2: i<=n

\_Đúng:

->thực hiện hành động lặp gán giá trị nb=nb\*i (nb có giá trị ban đầu là 1)

->gán i=i+2 và quay lại bước 2.1.2

\_sai: thoát câu lệnh 2.1

-2. 2: Nếu sai:

+2.2.1:cho i=1

+2.2.2: i<=n

\_ Đúng:

->thực hiện hành động lặp gán giá trị nb=nb\*i (nb có giá trị ban đầu là 1)

->gán i=i+2 và quay lại bước 2.2.2

\_Sai:thoát câu lệnh 2.2

-B3:hiện ra màn hình "n!! =" và giá trị: nb

d:) code

#include<iostream>

using namespace std;

int main()

{

int n,i;

int nb=1;

cout<<"nhap n:";

cin >>n;

if(n%2==0){

for(i=2;i<=n;i=i+2)

{nb=nb\*i;}}

else {

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

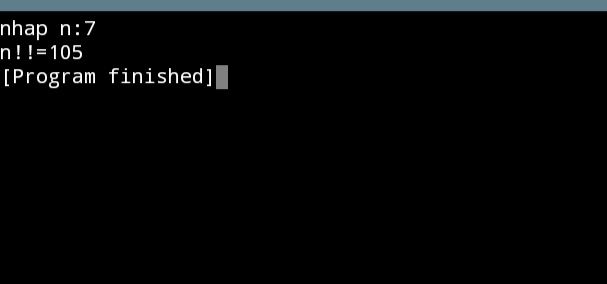
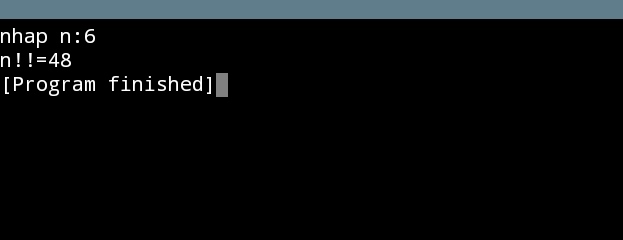
nb=nb\*i;}

cout<<"n!!="<<nb;

return 0;

}

e:) Test



**Bài 5:**

a:) input:nhập chuỗi kí tự

b:) output:cho kết quả (thống kê các số từ 1->9)

c:) process

-bẳt đầu:

-Bước1:khai báo char a[100]

-Bước 2: truyền <a> vào mảng void thongke(a)

\_b1: nhập chuỗi kí tự (vào mảng char a[100]) và hiện chuỗi

\_b2: chạy vòng lặp for(i=0;i<n;i++)

-2.1:cho i=0

-2.2: xét i<n

\_ Đúng:

+so sánh miền giá trị của từng kí với bảng mã ASCII với câu lệnh if và else if (kí tự 0->9 tương ứng

với mã từ 48->57, và dùng số nguyên dem0, dem1... dem9 tương ứng để đếm kí tự thỏa mãn khi xuất hiện )

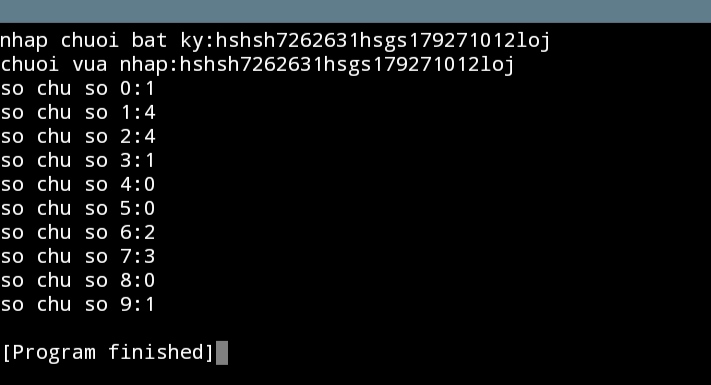
+tăng i++ và quay lại bước 2.2

\_Sai: thoát vòng lặp for

\_b3: hiện thống kê các số từ 0->9 theo lần lượt các dem0, dem1...dem9

-kết thúc chương trình

d:) test



**Bài 6:**

a:) input:nhập chuỗi họ tên

b:) output: hiện tên ra màn hình

c:) process

-Bắt đầu:(chương trình chỉ đúng khi xâu kí tự đã chuẩn hóa)

-Bước 1:khai báo char m[100] (mảng m gồm 100 kí tự)

-Bước 2: nhập chuỗi họ tên vào mảng m

-Bước 3: chuyền m vào hàm tách từ (câu lệnh như sau tachtu(m) ) vời hàm tachtu như sau:

\_b3.1:trong hàm tách từ thực hiện khai báo kiểu dữ liệu char với tham số hình thức m để

truyền tham số mảng m(hàm main) vào tham số hình thức, cho giá trị <i> bằng chiều dài chuỗi

m

\_b3.2:chạy vòng lặp while thực hiện giảm i đi 1 giá trị đến khi kí tự m[i-1] là khác cách hoặc

i<=0 thì dừng

\_b3.3: tiếp tục dùng while thực hiện gán giá trị m[i] vào tu[j] , tăng i và j thêm 1 và dừng câu

lệnh khi giá trị tham số <i> lớn hơn độ dài chuỗi m

\_b3.4: hiện ra màn hình "tên:" <<tu là tên của mình

\_Kết thúc chương trình

d:) code:

#include<iostream>

#include<string.h>

using namespace std;

void tachtu(char m[ ]){

int j=0;

int i=strlen(m);

char tu[30];

while(m[i-1]!=' ' && i>0){

i--;

}

while(i<=strlen(m)){

tu[j]=m[i];

i++;

j++;

}

cout<<"ten:"<<tu;

}

int main()

{char m[100];

cout<<"nhap ho ten:";

fflush(stdin);

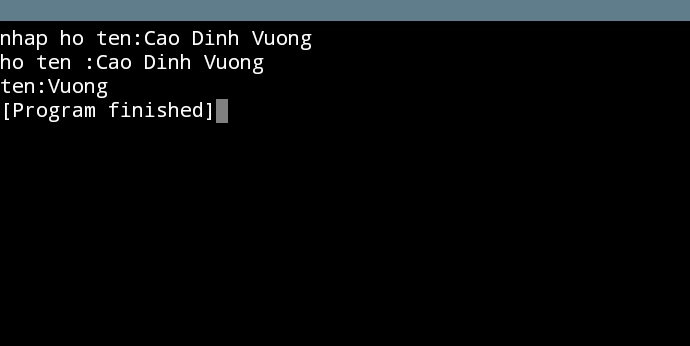
gets(m);

cout<<"ho ten :"<<m<<endl;

tachtu(m);

}

e:) Test



**Bài 7:**

a:) input: nhập mảng n số nguyên(0<=n<=50)

b:) output:hiện mảng, xóa phần tử vị trí k, xóa phần tử chia hết cho 3, đếm phần tử khác nhau

c:) process

\_bắt đầu

\_Bước 1: nhập n, nếu n<0 hoặc n>50 thì còn nhập lại n

\_Bước 2: khí có n ta thực hiện vòng lặp nhập giá trị m[i] với i bắt đầu bằng 0 (i=0) và i=i+1 đến khi

không còn thỏa mãn i<n thì dừng

\_Bước 3: hiện mảng

-ta cho i=0 chạy vòng lặp hiện giá trị m[i] , tăng i =i+1

-dừng vòng lặp khi i không thỏa mãn i<n

\_Bước 4: xóa vị trí thứ k

-nhập vị trí k

-cho i=0 chạy vòng lặp đến khi i không thỏa i<n

+nếu i = k ta giảm n=n-1 và thực hiện vòng lặp gán giá trị m[j]=m[j+1] , tăng j lên 1 đơn vị ( với j=i

)và dừng vòng lặp khi không thỏa mãn j<n) khi thoát vòng lặp tăng i thêm 1 và quay lại

+ i khác k tăng i lên thêm 1 (i++) và quay lại vòng lặp

-ta thực hiện hiện mảng như Bước 3 ( nếu muốn)

\_Bước 5: xóa phần tử chia hết cho 3

-cho i=0 , ta xét i<n

+Đúng : m[i] chia cho 3

-> nếu giá trị m[i] chia hết cho 3 ( tức chia 3 dư 0 ) ta thực hiện vòng lặp để gán giá trị m[j+1] cho m[i]

(Với j=i) và giảm n đi 1 , tăng j thêm 1 và quay lại vòng lặp thực hiện đến khi < j > lớn hơn hoặc bằng n thì

dừng ( lúc này ta được giá trị mảng m khác Mảng bạn đầu ) , quay lại xét điều kiện i<n và tiếp tục

-> còn không thì ta tăng i thêm 1 quay lại xét i<n và tiếp tục

+Sai: thoát và thực hiện câu lệnh tiếp.

-hiện mảng sau xóa ta thực hiện như Bước 3.

\_Bước 6: đếm phần tử ta cho i=0 thực hiện vòng lặp với điều kiện i<n và tăng i lên 1 khi thực xong câu

lệnh lặp

-Nếu i<n: thực hiện gán giá trị j=i+1, dùng vòng lặp while khi m[i] khác m[j] và thực hiện

+đầu tiên ta xét xem m[j] và m[i] có khác nhau không và j có bằng n-1 khôngnếu thỏa mãn ta

tăng biến giá trị <dem> thêm 1( với dem là số nguyên có giá trị ban đầu là 1).

+Tăng giá trị j thêm 1 khi và quay lại xét điều kiên i<n.

-khi i không thỏa mãn điều kiện ta thoát vòng lặp và thực hiện hiển thị số phần tử khác nhau trong

mảng =<dem>

d:) code

#include<iostream>

#include<iomanip>

using namespace std;

typedef int mang[100];

void nhap( int m[ ], int &n){

int i;

cout<<"nhap so phan tu n:";

cin>>n;

while(n>50 || n<0){

cout<<"nhap lai n:";

cin>>n;}

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

cout<<"m["<<i<<"]=";

cin>>m[i];}

cout<<"hien mang:";

}

void hien(int m[ ], int &n){

int i;

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

cout<<setw(3)<<m[i];

}

cout<<endl;

}

void xoak(int m[ ],int &n){

int i,j,k;

cout<<"nhap vi tri k:";

cin>>k;

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

if(i==k){

n--;

for(j=i;j<n;j++){

m[j]=m[j+1]; }

}}

cout<<"mang sau xoa vtri k:";

}

void xoa3(int m[ ],int &n){

int i=0,j;

while(i<n){

if(m[i]%3==0){

for(j=i;j<n;j++)

{m[j]=m[j+1];}

n--;}

else i++;}

cout<<"mang sau xoa ptu chia 3:";

}

void dempt(int m[ ],int &n){

int i;

int j;

int dem=1;

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

j=i+1;

while(m[i]!=m[j]){

if(m[j]!=m[i] && j==n-1){

dem++;}

j++;}}

cout<<"so phan tu khac nhau:"<<dem;

}

int main(){

mang a;

int na;

nhap(a,na);

hien(a,na);

xoak(a,na);

hien(a,na);

xoa3(a,na);

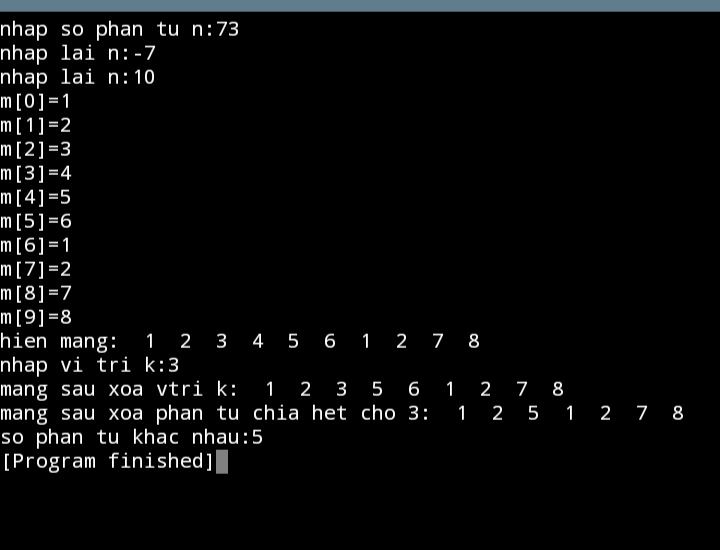
hien(a,na);

dempt(a,na);

return 0;

}

d:) Test:



**Bài 8:**

a:) Input: ma trận n\*m

b:) output: hiện ma trân, hàng phần tử âm nhiều nhất, vị trí chính phương, số phần từ bằng

phần tử vị trí (0, 0)

c:) process:

\_bắt đầu:

\_Bước 1: Nhập mảng

-1.1:nhập kích thước mảng n, m

-1.2:chạy vòng lặp for với i=0 và xét điều kiện i<n

+thỏa mãn:chạy tiếp vòng lặp for với j=0, nếu j<n thì thực hiện câu lệnh nhập phần từ h[i][j] và

tăng j thêm 1 quay lại xét điều kiện và tiếp tục đến khi không thỏa mãn j<n thì thoát vòng lặp trên, tăng i

thêm 1 và quay lại xét điều kiện i<n tiếp tục...

+không thỏa mãn i<n thoát vòng lặp

\_Bước 2: hiện mảng cho i=0 và điều kiện i<n

-thỏa mãn ta cho j=0 , xét điều kiện j<m nếu thỏa mãn thực hiện cậu lệnh hiện giá trị h[i][j] ,tăng j

thêm 1 và thực hiện lại câu lệnh đến khi j không thỏa mãn j<m thì tăng i thêm 1 quay lại xét điều kiện

i<n và tiếp tục lặp lại câu lệnh.

-không thỏa mãn thoát câu lệnh for

\_Bước 3: hiện hàng phần tử âm nhiều nhất

-ta thực hiện lệnh for lồng for và if : với mỗi giá trị i của vòng for ngoài ( i bắt đầu từ 0 đến n-1)

ta thực hiện :

+lệnh for trong với j ( j bắt đầu từ 0 đến m-1) với lệnh lặp đếm phần tử âm của hàng i bằng cách

xem giá trị h[i][j] nếu nhỏ hơn 0 thì ta tăng dem ( với dem ban đầu là 0)

+lệnh if trong để so sánh giá trị dem vừa có được ở lệnh for ở trên với dem1 ( dem1 ban đầu =0): nếu

dem>dem1 ta gán giá trị <dem> cho <dem1> , gán vtri=i, dem=0

-chạy for với j=0 và j chạy đến m-1 thực hiện câu lệnh để hiện giá trị h[vtri][j] hoàn thành for ta được hàng

phần tử âm nhiều nhất.

\_Bước 4: kiểm tra số chính phương:sử dụng 2 for lồng nhau với mỗi giá trị < i >vòng for ngoài ( i

chạy từ 0 đến n-1, i tăng thêm 1 sau câu lệnh lặp) ta thực hiện lệnh for trong với mỗi giá trị < j > ( j bắt đầu

từ 0 đến m-1) ta gán giá trị spr=sprt(h[i][j]) nói cách khác ta gán spr bằng căn bậc 2 của h[i][j] rồi mang

số sánh nếu giá trị của h[i][j] bằng bình phương của giá trị <spr> ta thực hiện câu lệnh tăng biến đếm

thêm 1 ( với dem là kiểu nguyên và có giá trị ban đầu bằng 0), tăng giá trị <j> thêm 1 sau câu lệnh lặp for

trong.

\_Bước 5: hiện vị trí phần tử là số chính phương ta sử dụng 2 for lồng nhau với mỗi giá trị < i >vòngfor ngoài ( i

chạy từ 0 đến n-1, i tăng thêm 1 sau câu lệnh lặp) ta thực hiện lệnh for trong với mỗi giá trị < j > ( j bắt đầu từ 0 đến m-

1) ta gán giá trị spr=sprt(h[i][j]) nói cách khác ta gán spr bằng căn bậc 2 của h[i][j] rồi mang so sánh nếu

giá trị của h[i][j] bằng bình phương của giá trị <spr> ta thực hiện câu lệnh xuất vị trí ra màn hình ( hay hiện

vị trí i, j), tăng giá trị <j> thêm 1 sau câu lệnh lặp for trong.

\_Bước 6: đếm phần từ =h[0][0] ta chạy 2 vòng for lồng nhau với mỗi giá trị biến i vòng for ngoài ( với i chạy từ

đến n-1 tăng i thêm 1 khi thực hiện xong câu lệnh lặp) ta hiện cậu lệnh lặp vòng for trong với mỗi giá trị j

( j chạy từ 0 đến m-1 và j tăng thêm 1 sau kết thúc câu lệnh lặp) ta thực hiện lệnh lặp so sánh giá trị h[i][j]

với h[0][0] nếu bằng nhau dem tăng thêm 1 ( với dem ban đầu có giá trị bằng -1) . Hiện giá trị biến dem ta

được số phần tử =h[i][j].

d:) Code

#include<iostream>

#include<iomanip>

using namespace std;

typedef int mang[100][100];

void nhap(int h[100][100] , int &n, int &m){

int i,j;

cout<<"nhap n:";

cin>>n;

cout<<"nhap m:";

cin>>m;

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

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

cout<<"h["<<i<<"]["<<j<<"]=";

cin>>h[i][j];}}

}

void hien(int h[100][100] , int &n, int &m){

int i,j;

cout<<"1.hien ma tran:"<<endl;

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

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

cout<<setw(5)<<h[i][j];}

cout<<endl;}

}

void hmam(int h[100][100] ,int &n,int&m){

int i,j,vtri;

int dem=0,dem1=0;

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

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

if(h[i][j]<0){

dem++;}

}

if(dem>dem1){

dem1=dem;

dem=0;

vtri=i;}

}

cout<<"2.hang co ptu am nhieu nhat:";

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

cout<<setw(3)<<h[vtri][j];}

}

void ktra(int h[100][100],int &n,int &m){

int i,j;

int sqr;

int dem=0;

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

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

sqr=sqrt(h[i][j]);

if(h[i][j]==sqr\*sqr){

dem++;

}

}

}

cout<<"\n3.so luong so chinh phuong trong ma tran:"<<dem;

}

void lietke(int h[100][100] ,int &n,int&m){

int i,j;

int sqr;

cout<<endl<<"4.vtri so chinh phuong:"<<endl;

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

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

sqr=sqrt(h[i][j]);

if(h[i][j]==sqr\*sqr){

cout<<"h["<<i<<"]["<<j<<"]"<<endl;} } }

}

void dem(int h[100][100],int &n ,int &m){

int i,j;

int dem=-1;

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

for(j=0;j<n;j++){

if(h[i][j]==h[0][0]){

dem++;}

} }

cout<<"5.so phan tu =a[0][0] la:"<<dem;

}

int main(){

mang a;

int n,m;

nhap(a,n,m);

hien(a,n,m);

hmam(a,n,m);

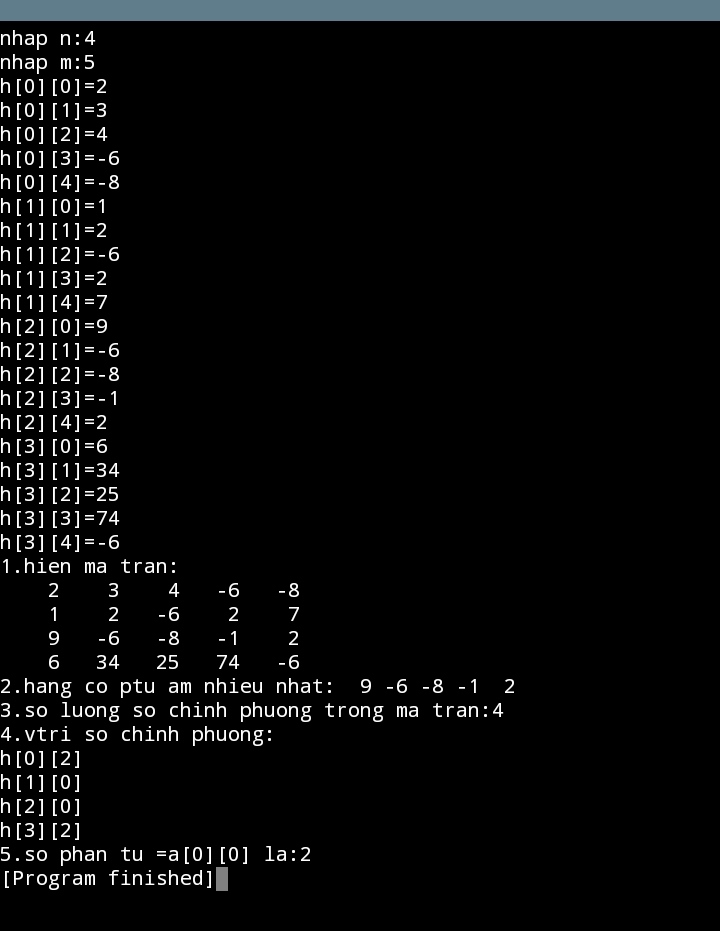
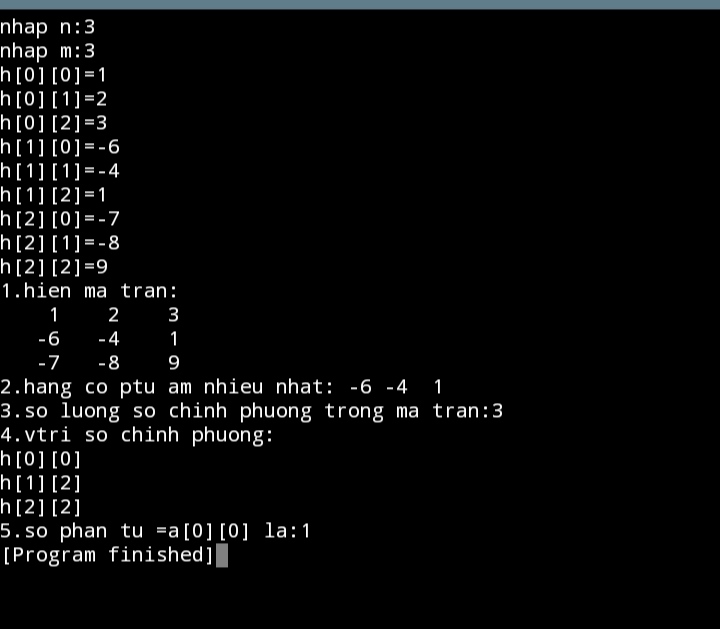
ktra(a,n,m);

lietke(a,n,m);

dem(a,n,m);

}

e:) Test



**Bài 9:**

a:) input: danh sách các sách

b:) output: chương trình con gồm

+nhâp-hiện danh sách,

+sắp xếp tến sách theo alphabet ghi vào file sach.txt

+sách của tác giả x in ở nhà suất bản y

c:) process

\_Bước 1: Xây dựng cấu trúc danh sách:sử dụng struct khai báo lần lượt các thành phần tên

sách, tên tác giả, nhà xuất bản theo kiểu kí tự <char> và năm xuất bản theo kiểu số nguyên.

\_Bước 2: Xây dựng các chương trình con

-bước 2.1: hàm nhập

+thực hiện khai báo các biến chương trình, gọi hàm struct,

+thực hiện nhập các thông tin :nhập số lượng danh sách <n> , dùng biến struct để nhập các

thông tin của từng thành phần của biến mỗi lần nhập một lượt các thông tin lưu vào phần tử thứ i

của mảng thì ta tăng phần

tử mảng thêm 1 cho đến khi phần tử mảng tiến đến n.

-bước 2.2: hàm hiện:khai báo cáo biến, gọi hàm struct,thực hiện xuất phần tử mảng của từng thành phần

struct ra màn hình đến khi hết mảng thì dừng .

-bước 2.3: hàm xếp danh sách theo tên sách và ghi vào file sach.txt.

+2.3.1:xếp danh sách ta thực hiện duyệt danh sách bằng cách sử dụng 2 vòng lặp for với mỗi i (i nguyên, i

chạy từ 0 tới n-1 và tăng i+1 sau câu lệnh lặp )vòng for ngoài ta thực hiện lệnh lặp vòng for trong với j

(nguyên, j chạy từ 0 đến n-1 và j+1 sau câu lệnh lặp) với lệnh lặp sử dụng câu lệnh so sánh từng giá trị

một trong phần tử j với phần tử j+1 nếu trả ra giá trị lớn hơn 0, tả đổi vị trí giá trị 2 phần tử cho nhau

duyệt đến khi hết phần tử của mảng

+2.3.2: thực hiện mở file "sach.txt" ở chế độ 'ios::out' với biến <ghitep>

->T.h1: nếu tệp được mở ta thực hiện duyệt danh sách từ đầu đến cuối từng danh sách vừa

xếp rồi lưu vào file "sach.txt" ( lưu bằng cách truyền dữ liệucủa từng phần tử trong mảng vào trong

file với câu lệnh ). Đóng file khi chạy xong. ở cách này file sẽ hiện dữ liệu truyền vào dạng kiểu char

nên khi check tệp ta sẽ thấy dữ liệu được truyền vào ở dạng text ( văn bản) < ở đây tả không thực hiện

check file vì đề bài nói không rõ ràng và em không kịp thơi gian :((> .

-> T.h2:Nếu không mở được tệp màn hình hiển thị " khong the mo file".

-bước 2.4: hàm thống kê : khai báo các biến của hàm ta thực hiện nhập vào tên tác giả 'x' bà nhà

xuất bản 'y'. Ta duyệt một lượt danh sách, lấy tên của từng tác giả và nhà suất bản mang so sánh nếu

tên và nhà suất bản lần lượt chùng với tên 'x' và nhà xuất bản 'y', thì tìm thấy danhsach[i] và hiện nó ra

màn hình , khi không tìm được thì không có giá trị trả và về màn hình không hiện danh sách.

-bước 2.5: hàm menu: thực hiện khai báo biến phù hợp ở đây tả dùng câu lệnh while để khi người

dúng nhập giá trị là 0 thì dừng hàm này.Trong lệnh while tả thiết kế với mỗi lựa chọn mặc định thì ta

hiện ra hàm con ứng với nội dung của nó. Khi nhập lựa chọn không nằm trong lựa chọn menu ta nhập

lại "n" , khi ta nhập 0 hoặc kiểu kí tự thì chương trình tự kết thúc.

\_Bước 3: ở hàm main thực hiện khai báo các biến, gọi hàm nhập và hàm menu( ở đây các

hàm còn lại sẽ được khai bao trong menu và hiện thông qua menu )

d:) code:

#include<iostream>

#include<string.h>

#include<iomanip>

#include<fstream>

using namespace std;

typedef struct{

char tensach[30];

char tentgia[30];

char nsban[100];

int namsb;}dulieu;

typedef dulieu m[100];

void nhap(dulieu a[ ],int &n){

int i;

cout<<"nhap so nguoi:";

cin>>n;

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

fflush(stdin);

cout<<" Nhap thong tin thu:"<<i+1<<endl;

cout<<"+nhap ten sach:";

gets(a[i].tensach);

cout<<"+nhap ten tac gia:";

gets(a[i].tentgia);

cout<<"+nha xuat ban:";

gets(a[i].nsban);

cout<<"+nam xuat ban:";

cin>>a[i].namsb;

}

}

void xuat(dulieu a[ ],int &n){

int i;

cout<<"xuat danh sach:"<<endl;

cout<<setw(15)<<"ten sach"<<setw(15)<<"ten tac gia"<<setw(15);

cout<<"nha xuat ban"<<setw(15)<<"nam xuat ban"<<endl;

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

cout<<setw(15)<<a[i].tensach<<setw(15);

cout<<a[i].tentgia<<setw(15);

cout<<a[i].nsban<<setw(15);

cout<<a[i].namsb<<endl;

}

}

void xep(dulieu a[ ],int &n){

int i,j;

int dem;

char t[20];

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

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

if (strcmp(a[j].tensach, a[j+1].tensach) > 0) {

strcpy(t, a[j].tensach);

strcpy(a[j].tensach, a[j+1].tensach);

strcpy(a[j+1].tensach, t);

strcpy(t, a[j].tentgia);

strcpy(a[j].tentgia, a[j+1].tentgia);

strcpy(a[j+1].tentgia, t);

strcpy(t, a[j].nsban);

strcpy(a[j].nsban, a[j+1].nsban);

strcpy(a[j+1].nsban, t);

dem=a[j].namsb;

a[j].namsb=a[j+1].namsb;

a[j+1].namsb=dem; }

}

}

fstream ghitep("sach.txt",ios::out );

if(ghitep.is\_open()){

cout<<"da ghi vao file vui long check file:";

ghitep<<"danh sach sach sau khi xep:";

ghitep<<"\n tsach"<<setw(15)<<"ten tgia"<<setw(15);

ghitep<<"nx ban"<<setw(15)<<"nam xuat ban"<<endl;

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

ghitep<<setw(10)<<a[i].tensach<<setw(15);

ghitep<<a[i].tentgia<<setw(15);

ghitep<<a[i].nsban<<setw(15);

ghitep<<a[i].namsb<<endl;

}

}

else {

cout<<"khong mo duoc fie de ghi";

}

ghitep.close();

}

void thongke(dulieu a[ ],int &n){

char x[30];

char y[100];

int i;

fflush(stdin);

cout<<"nhap tac gia X:";

gets(x);

fflush(stdin);

cout<<"nhap nha xuat ban Y:";

gets(y);

cout<<"sach cua tac gia X trong nsb Y"<<endl;

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

if(strcmp(a[i].tentgia,x)==0 && strcmp(a[i].nsban,y)==0){

cout<<a[i].tensach<<setw(15);

cout<<a[i].tentgia<<setw(15);

cout<<a[i].nsban<<setw(15);

cout<<a[i].namsb<<endl; }

}

}

void menu(dulieu a[ ],int &n){

int dem=1;

while(dem!=0){

cout<<"he thong menu:";

cout<<"\n1.hien danh sach:";

cout<<"\n2.xep danh sach theo chu cai";

cout<<"\n3.thong ke sach tac gia X in o nsb Y";

cout<<"\n0.thoat ctrinh";

cout<<"\nmoi nhap lua chon:";

cin>>dem;

if (dem==1){

xuat(a,n);

cout<<endl;

}

else if(dem==2){

xep(a,n);

cout<<endl;

}

else if(dem==3){

thongke(a,n);

cout<<endl;

}

else{cout<<"nhap lai lua chon hoac chon 0 de thoat:"<<endl;}

system("pause");

system("cls");

}

}

int main(){

dulieu a[100];

int n;

int dem=1;

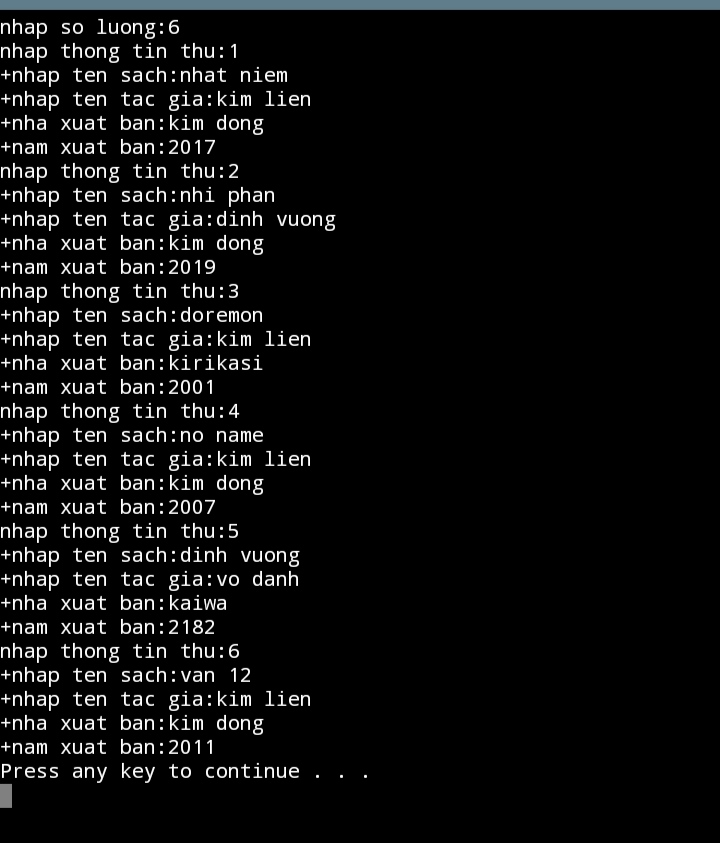
nhap(a,n);

system("pause");

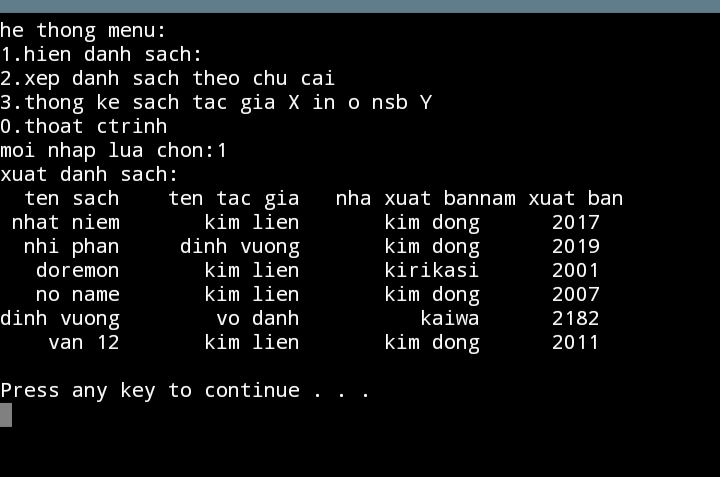
system("cls");

menu(a,n); }

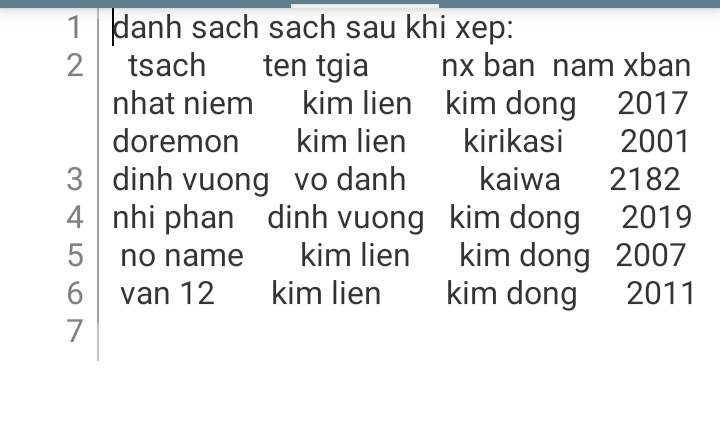
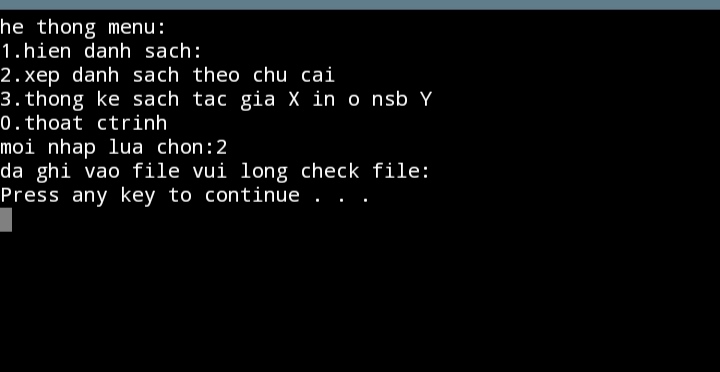
e:) Test nhập danh sách



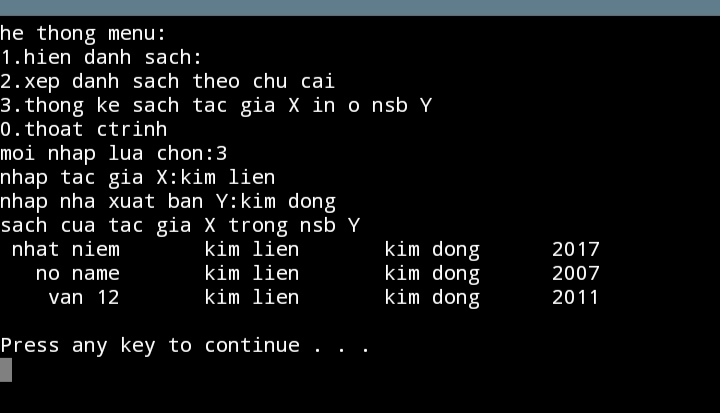
1: hiện danh sách



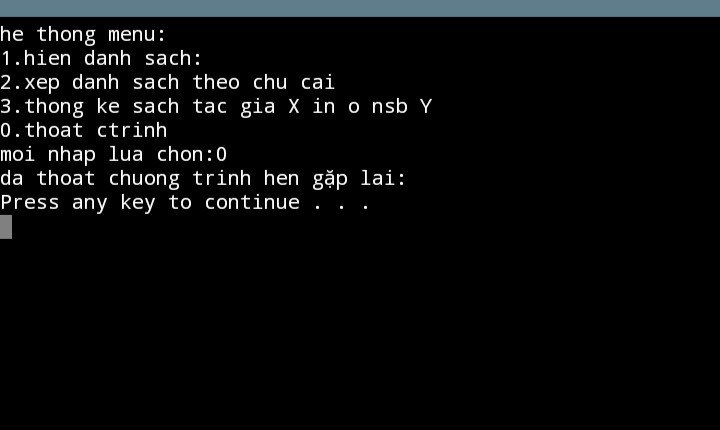
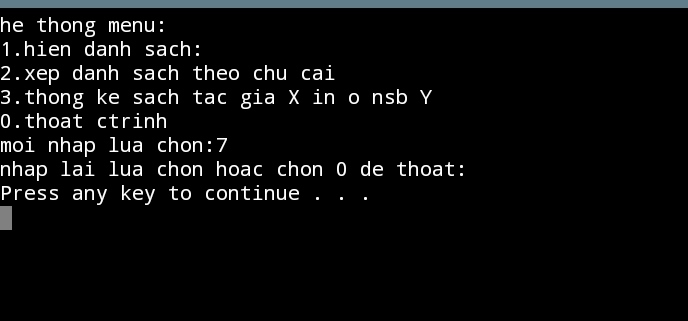
2: xếp danh sách và ghi vào file dữ liệu trong file ( uy tín)



3: Danh sách :sách của tác giả X trong nhà xuất bản Y



Khi ta nhập khác trong lựa chọn hoặc chọn 0



**Bài 10:**

a:) input: danh sách cán bộ

b:) output

+danh sách cán bộ

+danh sách cán bộ lương >5.0, ghi vào tệp "canbo.dat"(nhị phân) <hiện ra nếu cần>

+sắp xếp lương tăng dần<hiện ra nếu cần>

+số cán bộ có lương thỏa mãn điều kiện

c:) process

\_Bước 1: Xây dựng cấu trúc danh sách bằng struct như sau

typedef struct {

char ten[20];

char pban[20];

float hsluong;

float lcoban;

float luong;

}canbo;

\_Bước 2: xây dựng các chương trình con

-2.1:hàm nhập: thực hiện khai báo kiểu struct <tên biến> và các biến để thực hiện

+nhập số lượng n phần tử của danh sách

+chạy vòng lặp với i=0 chạy đến n-1 để tiến hành nhập câu lệnh lặp các thông tin với phần

tử thứ i của mảng tên, phòng ban... và lương được gán giá trị bằng công thức đã cho (lương=hs

lương\*lương cơ bản) tăng i

thêm 1 sau câu lệnh lặp.chương trình dừng hoặc không chạy khi i không thỏa mãn i<n.

-2.2:hiện danh sách:thực hiện khai báo các phần tử cần dùng

+ ta tiến hành chạy vòng lặp for với mỗi giá trị i bắt đầu từ 0

nếu thỏa mãn i<n ta tiến hành câu lệnh lặp hiện giá trị của mảng với phần tử thứ "i" lần lượt

tên,phòng ban, hệ số lương, lương cơ bản, lương và tăng i=i+1 sau câu lệnh lặp và quay lại xét điều

kiện.

+thoát câu lệnh khi i không thỏa mãn i<n ( hay đã chạy hết mảng n phần tử)

-2.3: hàm danh sách hệ số lương >5.0 và in vào file canbo.dat(dạng nhị phân)

+thực hiện mờ file "canbo.dat" dạng "ios::out" và "ios::binary" ( dạng ghi dữ liệu vào file và

dưới dạng nhị phân)

+duyệt mảng danh sách cán bộ với phần tử thứ i=0 (đến khi i không thỏa mãn i<n thì dừng)

:thực hiện câu lệnh lặp so sánh khi hệ số lương của phần tử thứ i nếu lớn hơn 5.0 ta hiện các thông

tin của mảng tại phần tử thứ [<i>] đồng thời truyền dữ liệu vào tệp "canbo.dat" (dạng nhị phân) , tăng

i=i+1 mỗi khi câu lệnh lặp kết thúc hoặc không thể thực hiện.

-2.4: hàm kiểm tra, lấy dữ liệu từ file "canbo.dat"để hiện ra man hình <nếu cần>.

+thực hiện khai báo phần tử cần thiết

+mở file " canbo.dat" ở chế độ "ios::in"( đọc dữ liệu từ file) và "ios::binary" (đọc/ghi dạng nhị

phân)

+Đọc lần lượt từng phần tử trong file và lưu vào mảng b[j], quá trình đọc liên tiếp đến khi hết

mảng thì thôi, mỗi lần đọc được một phần tử thì số phần tử <j> của mảng tăng lên.Sau khi kết thúc

đóng file

+để hiện mảng sau khi làm như sau:ta tiến hành chạy vòng lặp for với mỗi giá trị i bắt đầu từ

0 nếu thỏa mãn i<j ta tiến hành câu lệnh lặp hiện giá trị của mảng với phần tử b[i] lần lượt tên,phòng

ban, hệ số lương, lương cơ bản, lương và tăng i=i+1 sau câu lệnh lặp và quay lại xét điều kiện.Thoát

câu lệnh khi i không thỏa mãn i<n ( hay đã chạy hết mảng n phần tử) .

\_2.5: hàm sắp xếp danh sách lương tăng dần:

+khai báo biến sử dụng trong hàm: ( danhsach a[ ], int &n)

+tiến hành chạy 2 vòng for với mỗi giá trị i vòng for ngoài ta thực hiện lặp vòng lặp for với

giá trị j của vòng for trong ta lần lượt so sánh "lương" của biến a[j] với a[j+1] nếu lớn hơn tiến hành đổi

tất cả giá trị 2 phân tử cho nhau ( bằng lệnh swap(<biến 1>, <biến 2>) )và tăng j=j+1,cứ như vậy đến

khi j>=n thì dừng .Tăng i=i+1 khi kết thúc vòng for trong và dừng for ngoài khi i không thỏa mãn i<n.

kết thúc 2 vòng lặp tả được danh sách mới theo yêu cầu.

+hiện mảng sau xếp ta làm như bước 2.2 ( không cần khai báo lại)

\_2.6: hàm đếm số lượng và hiện danh sách cán bộ thỏa mãn điều kiện "T".

+khởi tạo hàm các biến tham số danhsach a[ ], int &n , dem=0, số nguyên t.

+nhập điều kiện t

+duyệt danh sách cán bộ nếu cán bộ nào có lương lớn hơn giá trị <t> ta tăng dem=dem+1

+thực hiện xuất giá trị <dem> ra màn hình ta đươc số lượng cán bộ có lương lớn hơn <t>

\_2.7: hàm menu

+khởi tạo hàm menu với kiểu trả về "void" ,kiểu dữ liệu fstream(file) , danhsach (struct), int

(nguyên) với lần lượt các biến (tự chọn)

+tạo biến <chon> kiểu số nguyên có giá trị bằng 1

+viết vòng lặp while khi giá trị biến <chọn> còn khác 0 thì chạy :trong vòng while tiến hành

hiện ra màn hình danh sách chọn lựa từ 0 đến 4. Ta tiến hành nhập giá trị cho biến <chon> , nếu giá trị

biến <chon> thuộc từ 0 đến 4 nếu khác màn hình hiển thị nhập lại <chon> đến khi thỏa mãn tiến

hành 1 trong các câu lệnh:

-> giá trị <chon> bằng 1 thực hiện hàm hiện ( bước 2.2)

-> giá trị <chon> bằng 2 thực hiện hàm xếp, hiện, lưu danh sách vào file những cán bộ có hệ

số lương >5.0 và hiện thử file ( bước 2.3 và 2.4)

-> chon=3 thực hiền hàm xếp danh sách ( bước 2.5)

-> chon=4 thực hiện đếm số cán bộ có lương thỏa mãn điều kiện "T" ( bước 2.6) .

Bước 3: khởi tạo hàm main

\_gọi , tạo các biến cho hàm main:fstream m ( gọi hàm file) ; canbo a[100] (hàm struct biến a);int

n ( hàm nguyên)

\_truyền dữ liệu vào các hàm: nhap(a, n) và menu(m, a, n)

d:) code

#include<iostream>

#include<iomanip>

#include<string.h>

#include<fstream>

using namespace std;

typedef struct{

char ten[20];

char pban[20];

float hsluong;

float lcoban;

float luong;

}canbo;

void nhap(canbo a[ ],int &n){

int i;

cout<<"nhap so luong can bo:";

cin>>n;

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

fflush(stdin);

cout<<"-ten can bo:";

gets(a[i].ten);

cout<<"phong ban:";

gets(a[i].pban);

cout<<"he so luong:";

cin>>a[i].hsluong;

cout<<"luong co ban:";

cin>>a[i].lcoban;

a[i].luong=a[i].lcoban\*a[i].hsluong;

}

}

void hien(canbo a[ ],int &n){

int i;

cout<<"hien danh sach nv:"<<endl;

cout<<setw(10)<<"ten"<<setw(12);

cout<<"p ban"<<setw(10)<<"hsluong"<<setw(10)<<"lco ban"<<setw(10)<<"luong"<<endl;

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

cout<<setw(10)<<a[i].ten<<setw(12);

cout<<a[i].pban<<setw(10);

cout<<a[i].hsluong<<setw(10);

cout<<a[i].lcoban<<setw(10);

cout<<a[i].luong<<endl;

}

}

void infile(fstream &m,canbo a[ ],int &n){

int i;

cout<<"can bo hs luong tren 5.0:"<<endl;

cout<<setw(10)<<"ten"<<setw(12);

cout<<"p ban"<<setw(10)<<"hs luong"<<setw(10)<<"lco ban"<<setw(10)<<"luong"<<endl;

m.open("canbo.dat",ios::out | ios::binary);

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

if(a[i].hsluong>5){

cout<<setw(10)<<a[i].ten<<setw(12);

cout<<a[i].pban<<setw(10);

cout<<a[i].hsluong<<setw(10);

cout<<a[i].lcoban<<setw(10);

cout<<a[i].luong<<endl;

m.write((char\*)&a[i],sizeof(canbo));

}

}

m.close();

}

void hienthu(canbo a[],int &n){

int j=0;

int i;

canbo b[100];

fstream m("canbo.dat",ios::in | ios::binary);

while(!m.eof()){

m.read((char\*)&b[j],sizeof(canbo));

j=j+1;

}

j=j-1;

m.close( );

cout<<"doc thu file:"<<endl;

for(i=0;i<j;i++){

cout<<setw(10)<<b[i].ten<<setw(12);

cout<<b[i].pban<<setw(10);

cout<<b[i].hsluong<<setw(10);

cout<<b[i].lcoban<<setw(10);

cout<<b[i].luong<<endl;

}

}

void xapxep(canbo a[],int&n){

int i,j;

char doi[100];

float dem;

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

for(j=i;j<n;j++){

if(a[j].luong>a[j+1].luong){

swap(a[j].ten,a[j+1].ten);

swap(a[j].pban,a[j].pban);

swap(a[j].hsluong,a[j+1].hsluong);

swap(a[j].lcoban,a[j+1].lcoban);

swap(a[j].luong,a[j+1].luong);

}

}

}

cout<<"mang sau sap xep:"<<endl;

cout<<setw(10)<<"ten"<<setw(12);

cout<<"pban"<<setw(10)<<"hsluong"<<setw(10)<<"lco ban"<<setw(10)<<"luong"<<endl;

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

cout<<setw(10)<<a[i].ten<<setw(12);

cout<<a[i].pban<<setw(10);

cout<<a[i].hsluong<<setw(10);

cout<<a[i].lcoban<<setw(10);

cout<<a[i].luong<<endl;

}

}

void demsl(canbo a[ ],int &n){

int i,dem=0;

int t;

cout<<"nhap dieu kien luong t:";

cin>>t;

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

if(a[i].luong>t){

dem++;

}

}

cout<<"so can bo co luong thoa man dieu kien:"<<dem;

}

void menu(fstream &m,canbo a[ ],int &n){

int chon=1;

cout<<"chao mung ban den b10:"<<endl;

while(chon!=0){

cout<<"menu lua chon:";

cout<<"\n1.hien danh sach can bo:";

cout<<"\n2.luu danh sach vao file va hien thu ds can bo hsluong >5:";

cout<<"\n3.xap xep danh sach va hien:";

cout<<"\n4.dem so luong can bo co luong thoa man dkien:";

cout<<"\n0.thoat chuong trinh:((";

cout<<"\nmoi nhap lua chon:";

cin>>chon;

system("cls");

if(chon==1){

hien(a,n);

cout<<endl;

}

else if(chon==2){

infile(m,a,n);

hienthu(a,n);

cout<<endl;

}

else if(chon==3){

xapxep(a,n);

cout<<endl;

}

else if(chon==4){

demsl(a,n);

cout<<endl;

}

else if(chon==0){

cout<<"thoat chuong trinh";

}

else{

cout<<"khong co lua chon nay nhap lai:"<<endl;

}

system("pause");

system("cls");

}

}

int main(){

fstream m;

canbo a[100];

int n,chon=1;

nhap(a,n);

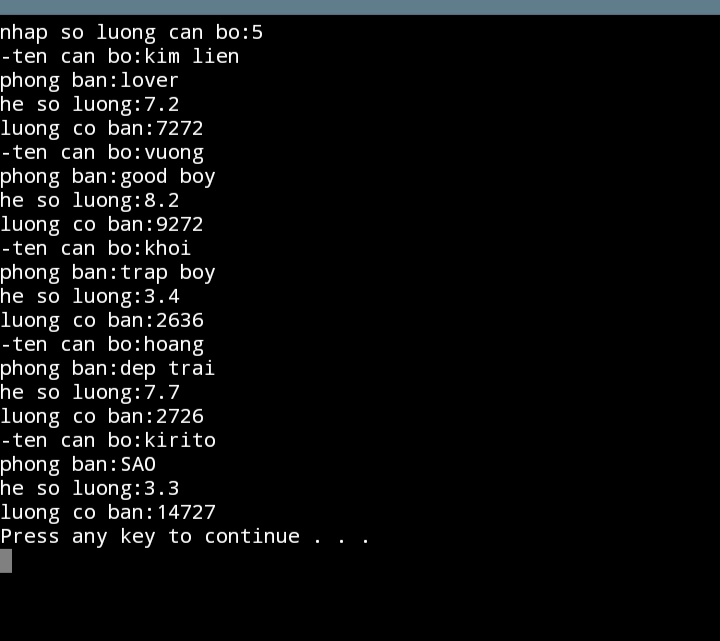
system("pause");

system("cls");

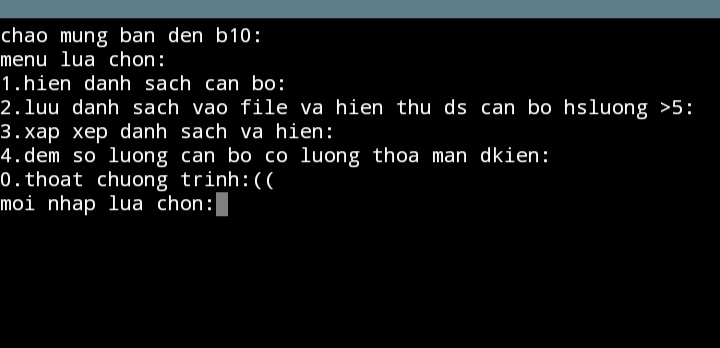
menu(m,a,n);

}

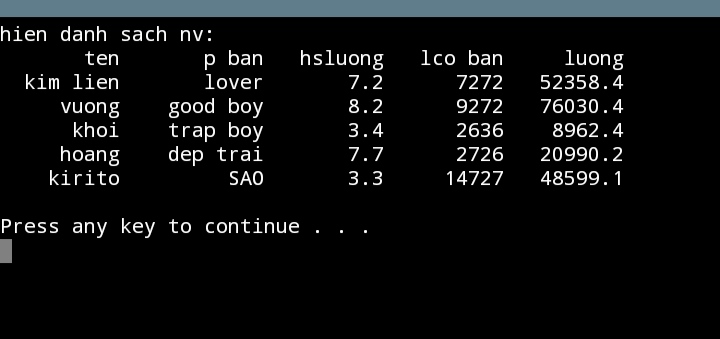
\_ Nhập danh sách cán bộ



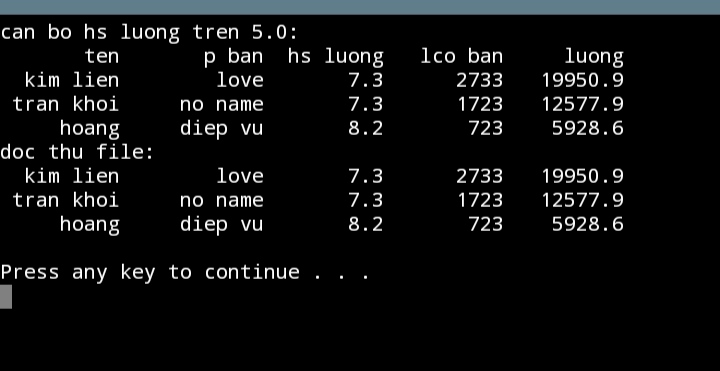
\_Menu:



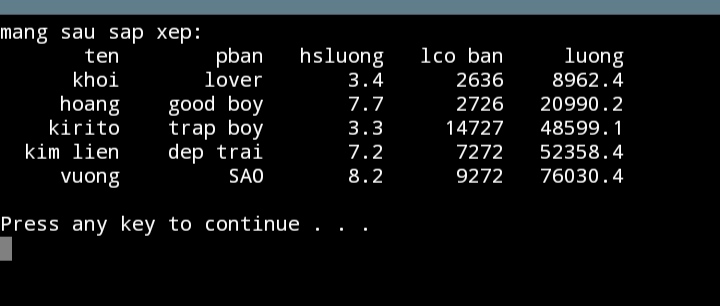
\_Hiện danh sách



\_ Danh sách cán bộ có hệ số lương trên 5.0 và hiện thử file



\_ Danh sách sắp xếp lương tăng dần



\_ Số lượng cán bộ có lương thỏa mãn điều kiện

