2001210004 – Lưu Đức Vinh

Bài về nhà Buổi 2:

Bài 1 và Bài 3:

#define \_CRT\_NONSTDC\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

void TaoMangNgauNhien\_PTlonhonbang15 (int \*&a, int &n);

void TaoMangNgauNhienSoChan (int \*&a, int &n);

void XuatM1C (int \*a, int n);

int Linear\_Search(int \*a, int n, int x);

void swap(int &x, int &y);

void TangDan\_InterchangeSort(int \*a, int n);

void GiamDan\_InterchangeSort(int \*a, int n);

int Binary\_Search (int \*a, int n, int x);

void Tangdan\_Selection\_Sort (int \*a, int n);

void Giamdan\_Selection\_Sort (int \*a, int n);

void Tangdan\_Quick\_Sort(int \*a, int left, int right);

void Giamdan\_Quick\_Sort(int \*a, int left, int right);

void LietKeSoNguyenTo\_NhohonN (int \*a, int n, int &x);

int TimChuSoDauLe (int n);

int kiemtrasnt (long n);

int TinhTongCacChuSoDauLe (int \*a, int n);

int DemSoLanXuatHien (int \*a, int n, int x);

void LietKeSoLanXuatHien\_CuaCacPT (int \*a, int n);

void SoLeDauMang\_SoChanCuoiMang (int \*a, int n);

void SoLeDauMang\_SoChanCuoiMang (int \*a, int n);

void Sapxep\_ChanTang\_LeGiam (int \*a, int n);

void TimMinThuHai (int \*a, int n);

void Sapxep\_ChanTangDan\_ConLaiGiuNguyenVitri (int \*a, int n);

int main ()

{

int \*a, n, chon;

do

{

printf("1.Tao mang 1 chieu ngau nhien co so pt >= 15\n");

printf("2.Tao mang chua toan so chan\n");

printf("3.Xuat mang\n");

printf("4.Tim kiem x trong mang theo giai thuat Linear Search\n");

printf("5.Sap xep mang tang dan/giam dan theo giai thuat Interchanger Sort\n");

printf("6.Tim kiem x trong mang theo giai thuat Binary Search\n");

printf("7.Sap xep mang tang dan/giam dan theo giai thuat Selection Sort\n");

printf("8.Sap xep mang tang dan/giam dan theo giai thuat Quick Sort\n");

printf("9.Liet ke so nguyen to nho hon n co trong mang\n");

printf("10.Tinh tong cac PT co chu so dau la le trong mang\n");

printf("11.Liet ke so lan xuat hien cua cac PT trong mang\n");

printf("12.Sap xep so le o dau mang, so chan o cuoi mang\n");

printf("13.Sap xep chan tang dan le giam dan\n");

printf("14.Tim PT nho thu 2 trong mang\n");

printf("15.Sap xep chan tang dan con lai giu nguyen vi tri\n");

printf("Chon chuc nang: "); scanf\_s("%d", &chon);

switch (chon)

{

case 1:

{

TaoMangNgauNhien\_PTlonhonbang15(a, n);

break;

}

case 2:

{

TaoMangNgauNhienSoChan(a, n);

break;

}

case 3:

{

XuatM1C(a, n);

break;

}

case 4:

{

int x;

printf("Nhap PT muon tim: "); scanf\_s("%d", &x);

if (Linear\_Search(a, n, x) == -1)

{

printf("Khong tim thay %d trong mang\n", x);

}

else

{

printf("Vi tri PT %d trong mang la: %d\n", x, Linear\_Search(a, n, x));

}

}

case 5:

{

int choose;

do

{

printf("\t1.Tang dan bang interchangesort\n");

printf("\t2.Giam dan bang interchangesort\n");

printf("\tChon chuc nang: "); scanf\_s("%d", &choose);

switch (choose)

{

case 1:

{

TangDan\_InterchangeSort(a, n);

break;

}

case 2:

{

GiamDan\_InterchangeSort(a, n);

break;

}

}

}

while (choose != 1 && choose != 2);

break;

}

case 6:

{

int x;

printf("\t\tVui long sap xep mang tang dan de dung Binary Search\n");

int choose;

do

{

printf("\t|1.Tang dan bang interchangesort\n");

printf("\t|Chon chuc nang: "); scanf\_s("%d", &choose);

switch (choose)

{

case 1:

{

TangDan\_InterchangeSort(a, n);

XuatM1C(a, n);

break;

}

}

}

while (choose != 1);

printf("Nhap PT muon tim: "); scanf\_s("%d", &x);

if (Binary\_Search(a, n, x) == -1)

{

printf("Khong ton tai gia tri %d\n", x);

}

else

{

printf("Vi tri PT %d trong mang la: %d\n", x, Binary\_Search(a, n, x));

}

break;

}

case 7:

{

int choose;

do

{

printf("\t1.Tang dan bang SelectionSort\n");

printf("\t2.Giam dan bang SelectionSort\n");

printf("\tChon chuc nang: "); scanf\_s("%d", &choose);

switch (choose)

{

case 1:

{

Tangdan\_Selection\_Sort(a, n);

break;

}

case 2:

{

Giamdan\_Selection\_Sort(a, n);

break;

}

}

}

while (choose != 1 && choose != 2);

break;

}

case 8:

{

int choose;

do

{

printf("\t1.Tang dan bang QuickSort\n");

printf("\t2.Giam dan bang QuickSort\n");

printf("\tChon chuc nang: "); scanf\_s("%d", &choose);

switch (choose)

{

case 1:

{

Tangdan\_Quick\_Sort(a, 0, n - 1);

break;

}

case 2:

{

Giamdan\_Quick\_Sort(a, 0, n - 1);

break;

}

}

}

while (choose != 1 && choose != 2);

break;

}

case 9:

{

int x;

LietKeSoNguyenTo\_NhohonN(a, n, x);

break;

}

case 10:

{

printf("Tong cac pt co chu so dau la le trong mang la: %d\n", TinhTongCacChuSoDauLe(a, n));

break;

}

case 11:

{

LietKeSoLanXuatHien\_CuaCacPT(a, n);

break;

}

case 12:

{

SoLeDauMang\_SoChanCuoiMang(a, n);

printf("Mang sau khi sap xep:\n");

XuatM1C(a, n);

break;

}

case 13:

{

Sapxep\_ChanTang\_LeGiam(a, n);

printf("Mang sau khi sap xep:\n");

XuatM1C(a, n);

break;

}

case 14:

{

TimMinThuHai(a, n);

break;

}

case 15:

{

Sapxep\_ChanTangDan\_ConLaiGiuNguyenVitri(a, n);

printf("Mang sau khi sap xep:\n");

XuatM1C(a, n);

break;

}

}

}

while (chon != 0);

return 0;

getch ();

}

void TaoMangNgauNhien\_PTlonhonbang15 (int \*&a, int &n)

{

printf("Nhap so luong pt: "); scanf\_s("%d", &n);

a = (int \*)malloc(n\*sizeof(int));

srand((unsigned)time(NULL));

if (n >= 15)

{

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

{

a[i] = rand() % 100;

}

}

else

{

printf("Xin vui long nhap so luong PT >= 15!\n");

}

}

void XuatM1C (int \*a, int n)

{

printf("Xuat mang:\n");

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

{

printf("%5d", a[i]);

}

printf("\n");

}

void TaoMangNgauNhienSoChan (int \*&a, int &n)

{

printf("Nhap so luong pt: "); scanf\_s("%d", &n);

a = (int \*)malloc(n\*sizeof(int));

srand((unsigned)time(NULL));

if (n > 0)

{

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

{

a[i] = (rand() % 100) \* 2;

}

}

else

{

printf("Xin vui long nhap so luong PT > 0!\n");

}

}

int Linear\_Search(int \*a, int n, int x)

{

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

{

if (a[i] == x)

return i;

}

return -1;

}

void swap(int &x, int &y)

{

int temp = x;

x = y;

y = temp;

}

void TangDan\_InterchangeSort(int \*a, int n)

{

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

{

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

{

if (a[j] < a[i])

{

swap(a[j], a[i]);

}

}

}

}

void GiamDan\_InterchangeSort(int \*a, int n)

{

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

{

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

{

if (a[j] > a[i])

{

swap(a[j], a[i]);

}

}

}

}

int Binary\_Search (int \*a, int n, int x)

{

int left = 0, right = n - 1;

while (left <= right)

{

int mid = (left + right) / 2;

if (a[mid] == x)

{

return mid;

}

else

{

if (a[mid] < x)

{

left = mid + 1;

}

else

{

right = mid - 1;

}

}

}

return -1;

}

void Tangdan\_Selection\_Sort (int \*a, int n)

{

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

{

int min = i;

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

{

if (a[j] < a[min])

{

min = j;

}

}

swap(a[i], a[min]);

}

}

void Giamdan\_Selection\_Sort (int \*a, int n)

{

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

{

int max = i;

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

{

if (a[j] > a[max])

{

max = j;

}

}

swap(a[i], a[max]);

}

}

void Tangdan\_Quick\_Sort(int \*a, int left, int right)

{

int l = left, r = right, mid = (left + right) / 2;

int x = a[mid];

do

{

while (a[l] < x) l++;

while (a[r] > x) r--;

if (l <= r)

{

swap(a[l], a[r]);

l++;

r--;

}

}

while (l <= r);

if (left < r) Tangdan\_Quick\_Sort(a, left, r);

if (right > l) Tangdan\_Quick\_Sort(a, l, right);

}

void Giamdan\_Quick\_Sort(int \*a, int left, int right)

{

int l = left, r = right, mid = (left + right) / 2;

int x = a[mid];

do

{

while (a[l] > x) l++;

while (a[r] < x) r--;

if (l <= r)

{

swap(a[l], a[r]);

l++;

r--;

}

}

while (l <= r);

if (left < r) Giamdan\_Quick\_Sort(a, left, r);

if (right > l) Giamdan\_Quick\_Sort(a, l, right);

}

int kiemtrasnt (long n)

{

int dem = 0;

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

{

if (n % i == 0)

{

dem++;

}

}

if (dem == 2)

{

return 1;

}

else

{

return 0;

}

}

void LietKeSoNguyenTo\_NhohonN (int \*a, int n, int &x)

{

int dem = 0;

do

{

printf("Nhap n: "); scanf\_s("%d", &x);

}

while (x < 1);

printf("Cac so nguyen to < %d co trong mang la:\n", x);

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

{

if (a[i] < x)

{

if (kiemtrasnt(a[i]) == 1)

{

printf("%5d", a[i]);

dem++;

}

}

}

printf("\n");

if (dem == 0)

{

printf("Mang ko co so nguyen to nao nho hon n!\n");

}

}

int TimChuSoDauLe (int n)

{

int b[1];

while (n > 0)

{

b[0] = n;

n = n / 10;

}

if (b[0] % 2 == 0)

{

return 0;

}

return 1;

}

int TinhTongCacChuSoDauLe (int \*a, int n)

{

int Tong = 0;

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

{

if (TimChuSoDauLe(a[i]) == 1)

{

Tong = Tong + a[i];

}

}

return Tong;

}

int DemSoLanXuatHien (int \*a, int n, int x)

{

int dem = 0;

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

{

if (a[i] == x)

{

dem++;

}

}

return dem;

}

void LietKeSoLanXuatHien\_CuaCacPT (int \*a, int n)

{

int \*b, dem = 0;

b = (int \*)malloc(n\*sizeof(int));

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

{

b[i] = DemSoLanXuatHien(a, n, a[i]);

}

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

{

printf("So lan xuat hien PT a[%d] = %d: %d\n", i, a[i], b[i]);

}

}

void Sapxep\_ChanTang\_LeGiam (int \*a, int n)

{

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

{

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

{

if (a[j] % 2 == 0)

{

swap(a[i], a[j]);

break;

}

}

}

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

{

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

{

if (a[i] % 2 == 0 && a[j] % 2 == 0)

{

if (a[i] > a[j])

{

swap(a[i], a[j]);

}

}

else if (a[i] % 2 == 0 && a[j] != 0)

{

break;

}

else

{

if (a[i] < a[j])

{

swap(a[i], a[j]);

}

}

}

}

}

void SoLeDauMang\_SoChanCuoiMang (int \*a, int n)

{

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

{

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

{

if (a[j] % 2 != 0)

{

swap(a[i], a[j]);

break;

}

}

}

}

void TimMinThuHai (int \*a, int n)

{

int Min\_Max = a[0];

int Min\_Min = a[0];

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

{

if (a[i] < Min\_Max)

{

Min\_Max = a[i];

}

}

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

{

if (a[i] < Min\_Min && a[i] != Min\_Max)

{

Min\_Min = a[i];

}

}

printf("PT nho thu hai trong mang la: %d\n", Min\_Min);

}

void Sapxep\_ChanTangDan\_ConLaiGiuNguyenVitri (int \*a, int n)

{

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

{

if (a[i] % 2 == 0)

{

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

{

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

{

swap(a[i], a[j]);

}

}

}

}

}

Bài 2 và Bài 3:

#define \_CRT\_NONSTDC\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

struct HonSo

{

int PhanNguyen;

int ts;

int ms;

};

struct PS

{

int ts;

int ms;

};

void Xuat1HS (HonSo \*hs);

void NhapN (int &n);

void Taodshs\_Random (HonSo \*dshs, int n);

void Xuatdshs (HonSo \*dshs, int n);

void SoSanh\_2HonSo (HonSo \*dshs, int n, int &vt1, int &vt2);

void xuat1PS (PS \*ps);

void ChuyenHonSo\_SangPhanSo (HonSo \*dshs, PS \*dsps, int n);

int USCLN (int x, int y);

void rutgon1PS (PS \*dsps);

void ChuyenPhanSo\_SangHonSo (HonSo \*dshs2, PS \*dsps, int n);

void TinhTong\_Hieu\_Tich\_Thuong\_2HonSo (HonSo \*dshs, PS \*dsps, int n, int &vt1, int &vt2);

HonSo RutGonHonSo(HonSo a);

HonSo TinhTong(HonSo a, HonSo b);

HonSo TinhHieu(HonSo a, HonSo b);

HonSo TinhTich(HonSo a, HonSo b);

HonSo TinhThuong(HonSo a, HonSo b);

void swap(HonSo &x, HonSo &y);

void TangDan\_InterchangeSort(HonSo \*dshs, PS \*dsps, int n);

void Vitri\_Max\_Min (HonSo \*dshs, PS \*dsps, int n);

void Vitri\_PhanNguyenChan (HonSo \*dshs, int n);

void TongcacPT (HonSo \*dshs, PS \*dsps, int n);

void XoatPT\_Vitrik (HonSo \*dshs, int &n, int &k);

void ThemHonSo (HonSo \*dshs, int &n, HonSo &x, int &k);

int main ()

{

int n, chon;

HonSo \*dshs;

PS \*dsps;

do

{

printf("1.Tao mang b chua gia tri hon so ngau nhien\n");

printf("2.Xuat danh sach hon so\n");

printf("3.So sanh 2 hon so\n");

printf("4.Chuyen hon so thanh phan so\n");

printf("5.Chuyen phan so thanh hon so\n");

printf("6.Tinh tong, hieu, tich, thuong 2 hon so\n");

printf("7.Sap xep b tang/giam theo 3 giai thuat sap xep\n");

printf("8.Tim va xuat vi tri PT Max/Min trong mang\n");

printf("9.Xuat vi tri PT co phan nguyen chan\n");

printf("10.Tong cac PT trong mang\n");

printf("11.Xoa pt thu k trong mang\n");

printf("12.Them hon so x tai vi tri k\n");

printf("Chon chuc nang: "); scanf\_s("%d", &chon);

switch (chon)

{

case 1:

{

NhapN(n);

dshs = (HonSo \*)malloc(n\*sizeof(HonSo));

Taodshs\_Random(dshs, n);

break;

}

case 2:

{

Xuatdshs(dshs, n);

break;

}

case 3:

{

int vt1, vt2;

SoSanh\_2HonSo(dshs, n, vt1, vt2);

break;

}

case 4:

{

dsps = (PS \*)malloc(n\*sizeof(PS));

ChuyenHonSo\_SangPhanSo(dshs, dsps, n);

break;

}

case 5:

{

HonSo \*dshs2;

dshs2 = (HonSo \*)malloc(n\*sizeof(HonSo));

ChuyenPhanSo\_SangHonSo(dshs2, dsps, n);

break;

}

case 6:

{

int vt1, vt2;

printf("Nhap 2 vi tri can tinh\n");

printf("Hon so vi tri: "); scanf\_s("%d", &vt1);

printf("Hon so vi tri: "); scanf\_s("%d", &vt2);

printf("Tong 2 hon so: ");

Xuat1HS(&TinhTong(\*(dshs + vt1), \*(dshs + vt2)));

printf("Hieu 2 hon so: ");

Xuat1HS(&TinhHieu(\*(dshs + vt1), \*(dshs + vt2)));

printf("Tich 2 hon so: ");

Xuat1HS(&TinhTich(\*(dshs + vt1), \*(dshs + vt2)));

printf("Thuong 2 hon so: ");

Xuat1HS(&TinhThuong(\*(dshs + vt1), \*(dshs + vt2)));

break;

}

case 7:

{

dsps = (PS \*)malloc(n\*sizeof(PS));

TangDan\_InterchangeSort(dshs, dsps, n);

break;

}

case 8:

{

dsps = (PS \*)malloc(n\*sizeof(PS));

Vitri\_Max\_Min(dshs, dsps, n);

break;

}

case 9:

{

Vitri\_PhanNguyenChan(dshs, n);

break;

}

case 10:

{

dsps = (PS \*)malloc(n\*sizeof(PS));

TongcacPT(dshs, dsps, n);

break;

}

case 11:

{

int k;

XoatPT\_Vitrik(dshs, n, k);

break;

}

case 12:

{

HonSo x;

int k;

ThemHonSo(dshs, n, x, k);

break;

}

}

}

while (chon != 0);

return 0;

getch ();

}

void Xuat1HS (HonSo \*hs)

{

printf("\tHon so: %d %d/%d\n", hs->PhanNguyen, hs->ts, hs->ms);

}

void NhapN (int &n)

{

printf("\tNhap danh sach hon so\n");

printf("Nhap so luong hon so: "); scanf\_s("%d", &n);

}

void Taodshs\_Random (HonSo \*dshs, int n)

{

srand((unsigned)time(NULL));

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

{

(dshs + i)->PhanNguyen = rand() % 100;

(dshs + i)->ts = rand() % 100;

(dshs + i)->ms = rand() % 100 + 1;

}

}

void Xuatdshs (HonSo \*dshs, int n)

{

printf("Danh sach hon so vua nhap la:\n");

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

{

printf("- Hon so thu %d:\n", i + 1);

Xuat1HS(dshs + i);

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

}

}

void SoSanh\_2HonSo (HonSo \*dshs, int n, int &vt1, int &vt2)

{

long double a[2];

printf("Nhap vi tri hon so muon so sanh thu 1: "); scanf\_s("%d", &vt1);

printf("Nhap vi tri hon so muon so sanh thu 2: "); scanf\_s("%d", &vt2);

a[0] = (float)(((dshs + vt1)->PhanNguyen \* (dshs + vt1)->ms) + (dshs + vt1)->ts) / (dshs + vt1)->ms;

a[1] = (float)(((dshs + vt2)->PhanNguyen \* (dshs + vt2)->ms) + (dshs + vt2)->ts) / (dshs + vt2)->ms;

if (a[0] > a[1])

{

printf("Hon so: %d %d/%d > %d %d/%d\n", (dshs + vt1)->PhanNguyen, (dshs + vt1)->ts, (dshs + vt1)->ms, (dshs + vt2)->PhanNguyen, (dshs + vt2)->ts, (dshs + vt2)->ms);

}

else if (a[0] < a[1])

{

printf("Hon so: %d %d/%d < %d %d/%d\n", (dshs + vt1)->PhanNguyen, (dshs + vt1)->ts, (dshs + vt1)->ms, (dshs + vt2)->PhanNguyen, (dshs + vt2)->ts, (dshs + vt2)->ms);

}

else

{

printf("Hon so: %d %d/%d = %d %d/%d\n", (dshs + vt1)->PhanNguyen, (dshs + vt1)->ts, (dshs + vt1)->ms, (dshs + vt2)->PhanNguyen, (dshs + vt2)->ts, (dshs + vt2)->ms);

}

}

void xuat1PS (PS \*ps)

{

if (ps->ts < 0 && ps->ms < 0)

{

printf("\tPhan so: %d/%d\n", ps->ts \* -1, ps->ms \* -1);

}

else if (ps->ms < 0)

{

printf("\tPhan so: %d/%d\n", ps->ts \* -1, ps->ms \* -1);

}

else if (ps->ms == 1)

{

printf("\tPhan so: %d\n", ps->ts);

}

else

printf("\tPhan so: %d/%d\n", ps->ts, ps->ms);

}

int USCLN (int x, int y)

{

if (x == 0) return y;

return USCLN(y % x, x);

}

void rutgon1PS (PS \*dsps)

{

int u = USCLN(dsps->ts, dsps->ms);

dsps->ts = dsps->ts / u;

dsps->ms = dsps->ms / u;

}

void ChuyenHonSo\_SangPhanSo (HonSo \*dshs, PS \*dsps, int n)

{

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

{

(dsps + i)->ts = ((dshs + i)->PhanNguyen \* (dshs + i)->ms) + (dshs + i)->ts;

(dsps + i)->ms = (dshs + i)->ms;

}

printf("Hon so sau khi chuyen thanh phan so\n");

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

{

rutgon1PS(dsps + i);

xuat1PS(dsps + i);

}

}

void ChuyenPhanSo\_SangHonSo (HonSo \*dshs2, PS \*dsps, int n)

{

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

{

(dshs2 + i)->PhanNguyen = (dsps + i)->ts / (dsps + i)->ms;

(dshs2 + i)->ts = (dsps + i)->ts % (dsps + i)->ms;

(dshs2 + i)->ms = (dsps + i)->ms;

}

printf("Phan so sau khi chuyen thanh hon so\n");

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

{

Xuat1HS(dshs2 + i);

}

}

HonSo RutGonHonSo(HonSo a)

{

if (a.ts > a.ms)

{

int tam = a.ts / a.ms;

a.PhanNguyen += tam;

a.ts -= a.ms \* tam;

}

int UCLN = USCLN(a.ts, a.ms);

a.ts /= UCLN;

a.ms /= UCLN;

return a;

}

HonSo TinhTong(HonSo a, HonSo b)

{

HonSo c;

c.PhanNguyen = a.PhanNguyen + b.PhanNguyen;

c.ts = a.ts \* b.ms + a.ms \* b.ts;

c.ms = a.ms \* b.ms;

return c;

}

HonSo TinhHieu(HonSo a, HonSo b)

{

HonSo c;

c.PhanNguyen = a.PhanNguyen - b.PhanNguyen;

c.ts = a.ts \* b.ms - a.ms \* b.ts;

c.ms = a.ms \* b.ms;

return c;

}

HonSo TinhTich(HonSo a, HonSo b)

{

a.ts = a.ts + a.PhanNguyen \* a.ms;

b.ts = b.ts + b.PhanNguyen \* b.ms;

HonSo c;

c.PhanNguyen = 0;

c.ts = a.ts \* b.ts;

c.ms = a.ms \* b.ms;

c = RutGonHonSo(c);

return c;

}

HonSo TinhThuong(HonSo a, HonSo b)

{

a.ts = a.ts+ a.PhanNguyen \* a.ms;

b.ts = b.ts + b.PhanNguyen \* b.ms;

HonSo c;

c.PhanNguyen = 0;

c.ts = a.ts \* b.ms;

c.ms = a.ms \* b.ts;

c = RutGonHonSo(c);

return c;

}

void swap(HonSo &x, HonSo &y)

{

HonSo temp = x;

x = y;

y = temp;

}

void TangDan\_InterchangeSort(HonSo \*dshs, PS \*dsps, int n)

{

float \*a;

a = (float \*)malloc(n\*sizeof(float));

ChuyenHonSo\_SangPhanSo(dshs, dsps, n);

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

{

a[i] = (float)(dsps + i)->ts / (dsps + i)->ms;

}

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

{

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

{

if ((float)a[j] < (float)a[i])

{

swap(\*(dshs + j), \*(dshs + i));

}

}

}

}

void Vitri\_Max\_Min (HonSo \*dshs, PS \*dsps, int n)

{

float \*a;

a = (float \*)malloc(n\*sizeof(float));

ChuyenHonSo\_SangPhanSo(dshs, dsps, n);

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

{

a[i] = (float)(dsps + i)->ts / (dsps + i)->ms;

}

float Min = a[0];

float Max = a[0];

int vtMax, vtMin;

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

{

if (a[i] > Max)

{

Max = a[i];

vtMax = i;

}

}

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

{

if (a[i] < Min)

{

Min = a[i];

vtMin = i;

}

}

printf("PT Max co gia tri %d %d/%d va dia chi o nho la %x\nPT Min co gia tri %d %d/%d va dia chi o nho la %x\n", (dshs + vtMax)->PhanNguyen, (dshs + vtMax)->ts, (dshs + vtMax)->ms, (dshs + vtMax), (dshs + vtMin)->PhanNguyen, (dshs + vtMin)->ts, (dshs + vtMin)->ms, (dshs + vtMin));

}

void Vitri\_PhanNguyenChan (HonSo \*dshs, int n)

{

printf("Xuat vi tri cac PT co phan nguyen chan\n");

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

{

if ((dshs + i)->PhanNguyen % 2 == 0)

{

printf("Vi tri PT %d %d/%d la: %x\n", (dshs + i)->PhanNguyen, (dshs + i)->ts, (dshs + i)->ms, (dshs + i));

}

}

printf("\n");

}

void TongcacPT (HonSo \*dshs, PS \*dsps, int n)

{

float \*a, Tong = 0;;

a = (float \*)malloc(n\*sizeof(float));

ChuyenHonSo\_SangPhanSo(dshs, dsps, n);

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

{

a[i] = (float)(dsps + i)->ts / (dsps + i)->ms;

}

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

{

Tong += a[i];

}

printf("Tong cac PT trong mang la: %.3f\n", Tong);

}

void XoatPT\_Vitrik (HonSo \*dshs, int &n, int &k)

{

do

{

printf("Nhap vi tri k muon xoa: "); scanf\_s("%d", &k);

}

while (k < n - 1 && k >= 0);

if (k < n - 1)

{

for (int i = k; i < n; i++)

{

dshs[i] = dshs[i + 1];

}

n--;

}

else

{

n--;

}

}

void ThemHonSo (HonSo \*dshs, int &n, HonSo &x, int &k)

{

printf("Nhap hon so x:\n");

printf("Nhap phan nguyen: "); scanf\_s("%d", &x.PhanNguyen);

printf("Nhap phan tu so: "); scanf\_s("%d", &x.ts);

printf("Nhap phan mau so: "); scanf\_s("%d", &x.ms);

printf("Nhap vi tri muon them: "); scanf\_s("%d", &k);

if (k < 0)

{

for (int i = n - 1; i >= k; i--)

{

dshs[i + 1] = dshs[i];

}

dshs[0] = x;

n++;

}

else if (k > n)

{

dshs[n] = x;

n++;

}

else

{

for (int i = n - 1; i >= k; i--)

{

dshs[i + 1] = dshs[i];

}

dshs[k] = x;

n++;

}

}