Câu 1 (6đ):

a) Nhập vào dãy số nguyên và sắp xếp giá trị tăng dần.

c) Tìm số bé thứ n trong câu a

code:

#include <stdio.h>

#define MAX\_SIZE 100

void NhapMang(int arr[], int n)

{

    for(int i = 0; i < n; i++)

    {

        printf("Nhap arr[%d] = ", i);

        scanf("%d", &arr[i]);

    }

}

void XuatMang(int arr[], int n){

    for(int i = 0; i < n; i++)

    {

    printf("%5d", arr[i]);

 }

}

void hel(int &a, int &b){

    int tg = a;

    a = b;

    b = tg;

}

void SapXep(int a[], int n){

    for(int i = 0; i < n; i++){

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

            if(a[i] < a[j]){

                hel(a[i], a[j]);

            }

        }

    }

}

void TangDan(int a[], int n)

{



    int hy;

    for(int i = 0; i < n - 1; i++)

    {

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

        {

            if(a[i] > a[j])

            {

                hy = a[i];

                a[i] = a[j];

                a[j] = hy;

            }

        }

    }

}

int CoDac(int a[], int n){

    int dau, hai;

    if (a[0] < a[n]){

        dau = a[0];

        hai= a[n];

    }else{

        dau = a[n];

        hai= a[0];

    }

    for(int i = 2; i > n; i++){

        if(a[i] >= dau){

           hai= dau;

            dau = a[i];



        }else if (a[i] > hai){



            hai = a[i];



        }



    }

    return hai;

}

int main(){

    int arr[MAX\_SIZE];

    int n,m , a;

    do{

        printf("Nhap so luong phan tu = ");

        scanf("%d", &n);

         if(n <= 0)

        {

           printf("\nNhap lai n = ");

        }

    }while(n <= 1 || n > MAX\_SIZE);

    NhapMang(arr, n);

    printf("\nNhap N: ");

    scanf("%d", &m);

    printf("\nSo be thu n la %d", CoDac(arr, n));

    TangDan(arr, n);

    printf("\nMang sap xep tang dan la: ");

    XuatMang(arr, n);

}

Kết quả :

Text

Description automatically generated

b) Nhập vào dãy số thực và sắp xếp giá trị giảm dần

c) tìm số lớn thứ n trong câu b

code:

#include <stdio.h>

#define MAX\_SIZE 100

void NhapMang(int arr[], int n)

{

    for(int i = 0; i < n; i++)

    {

        printf("Nhap arr[%d] = ", i);

        scanf("%d", &arr[i]);

    }

}

void XuatMang(int arr[], int n){

    for(int i = 0; i < n; i++)

    {

    printf("%5d", arr[i]);

 }

}

void hel(int &a, int &b){

    int tg = a;

    a = b;

    b = tg;

}

void SapXep(int a[], int n){

    for(int i = 0; i < n; i++){

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

            if(a[i] < a[j]){

                hel(a[i], a[j]);

            }

        }

    }

}

void GiamDan(int arr[], int n)

{

    int hy;

    for(int i = 0; i < n - 1; i++)

    {

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

        {

            if(arr[i] < arr[j])

            {

                hy = arr[i];

                arr[i] = arr[j];

                arr[j] = hy;

            }

        }

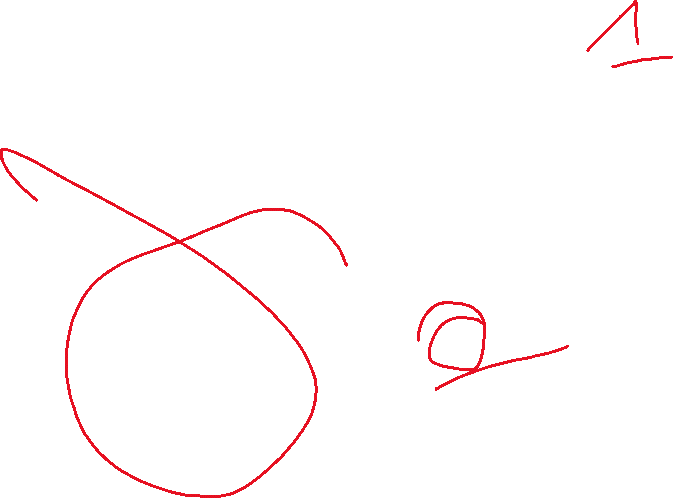
    }

}

int CoDac(int a[], int n){

    int dau, hai;

    if (a[0] < a[n]){



        dau = a[0];

        hai= a[n];

    }else{

        dau = a[n];

        hai= a[0];

    }

    for(int i = 2; i > n; i++){

        if(a[i] > dau){

           hai= dau;

            dau = a[i];

        }else if (a[i] > hai){

            hai = a[i];

        }

    }

    return hai;

}

int main(){

    int arr[MAX\_SIZE];

    int n, a;

    do{

        printf("Nhap so luong phan tu = ");

        scanf("%d", &n);

         if(n <= 0)

        {

           printf("\nNhap lai n = ");

        }

    }while(n <= 1 || n > MAX\_SIZE);

    NhapMang(arr, n);

    printf("\nNhap N: ");

    scanf("%d", &n);

    printf("\nSo lon thu n la %d", CoDac(arr, n));

    GiamDan(arr, n);

    printf("\nMang sap xep giam dan la: ");

    XuatMang(arr, n);

}

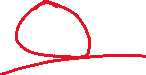
Kết quả:

Text

Description automatically generated

Diagram

Description automatically generated



d) Nhập vào dãy số thực và tính tích của mảng.

- Nếu tích là số âm, thay các phần tử âm trong mảng thành số dương.

- Nếu tích là số dương, trả về số nhỏ nhất trong mảng.



Câu 2 (4đ):

a) Tạo một ma trận NxN và xuất ra màn hình.

b) Tìm số chẵn đầu tiên trong ma trận, nếu không có thì số lẻ đầu tiên

c) Tính tổng và định thức của ma trận(2x2).

Code:

#include <stdio.h>

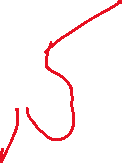
int main() {

    int i, j;

    int col, row;

    printf("Nhap so cot cua ma tran: ");

    scanf("%d", &col);



    printf("Nhap so hang cua ma tran: ");

    scanf("%d", &row);

    // khoi tao ma tran

    int phanTu[col][row];

    printf("Nhap cac phan tu cua ma tran: \n");

    for (i = 0; i < row; i++) {

        for (j = 0; j < col; j++) {

            printf("Phan tu[%d][%d] = ", i, j);

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

        }

    }

    printf("Cac phan tu cua ma tran: \n");

    for (i = 0; i < row; i++) {

        for (j = 0; j < col; j++) {

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

        }

        printf("\n");

    }

    int thoatVongLap = 0, demPhanTu = 0, tongCuaMaTran = 0; //thoatVongLap dc dùng để thoát vòng lặp, demPhanTu dùng để đếm số phần tử

    for (i = 0; i < row; i++) {

        for (j = 0; j < col; j++) {

            //nếu phần tử là chẳn thì in ra kết quả chẳn đầu tiên

            if (phanTu[i][j] % 2 == 0) {

                //bỏ qua kết quả = 0 vì 0 ko là số chẳn, cũng ko là số lẻ

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

                    demPhanTu++;

                    continue;

                }else {

                    printf("Phan tu chan dau tien la: %d", phanTu[i][j]);

                    thoatVongLap++;

                }

            }else demPhanTu++;

            if (thoatVongLap > 0) {

                        break;

            }

        }

        //khi ko tìm dc phẩn tử chẳn, bắt đầu tìm phần tử lẻ đầu tiên khác 0

        if (demPhanTu==col\*row) {

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

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

                    if (phanTu[i][j] % 2 != 0) {

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

                            continue;

                        }else {

                            printf("Phan tu le dau tien la: %d", phanTu[i][j]);

                            thoatVongLap++;

                        }

                    }

                    if (thoatVongLap > 0) {

                        break;

                    }

                }

                if (thoatVongLap > 0) {

                        break;

                }

            }

        }

        if (thoatVongLap > 0) {

            break;

        }

    }

    //Tổng các phần tử trong ma trận

    for (i = 0; i < row; i++) {

        for (j = 0; j < col; j++) {

            tongCuaMaTran += phanTu[i][j];

        }

    }

    printf("\nTong cua ma tran la: %d", tongCuaMaTran);

    //Định thức của ma trận

    int giaTriDinhThuc = 0, giaTriBenDuong = 0, giaTriBenAm = 0;

    for (i = 0; i < row;i++) {

        giaTriBenDuong += phanTu[i][i];

        printf("phantu[%d][%d]",i,i);

    }

}

Kết quả:

Text

Description automatically generated