Họ và tên: Lê Quang Anh Ngày: 22/12/2022

Git: [QuangAnh6723/Test\_CLB (github.com)](https://github.com/QuangAnh6723/Test_CLB)

Buổi 2 C

Câu 1 (6đ):

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

#include<stdio.h>

#define SIZE(arr) (sizeof(arr) / sizeof(arr[0]))

void nhap(int arr[], int size)

{

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

    for (int i = 0; i < size; i++)

    {

        printf("Nhap phan tu thu %d: ", i);

        scanf("%d", &arr[i]);

    }

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

}

void xuat(int arr[], int size)

{

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

    for(int i = 0; i < size; i++ )

    {

        printf("%d ", arr[i]);

    }

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

}

void swap(int \*a, int \*b)

{

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void tangDan(int arr[], int size)

{

    for (int  i = 0; i < size; i++)

    {

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

        {

            if(arr[i] > arr[j])

            {

                swap(&arr[i], &arr[j]);

            }

        }

    }

}

int main()

{

    // int a[] = {10, 10, 10, 9, 8, 6, 15, 20, 30, 26, 1, 6 , 7};

    int a[10] = {0};

    printf("\n\nNhap mang int: \r\n");

    nhap(a, SIZE(a));

    printf("Mang chua duoc sap xep la: \r\n");

    xuat(a, SIZE(a));

    printf("Mang da dc sap xep tang dan la: \r\n");

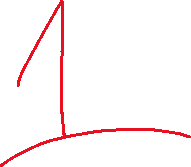
    tangDan(a, SIZE(a));

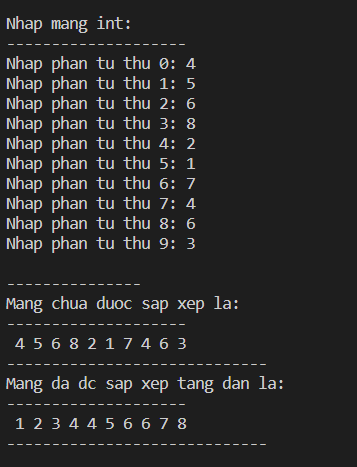
    xuat(a, SIZE(a));

    printf("\n");

    return 0;

}





1. Nhập vào dãy số thực và sắp xếp giá trị giảm dần.

#include<stdio.h>

#define SIZE(arr) (sizeof(arr) / sizeof(arr[0]))

void nhap(float arr[], int size)

{

    printf("---------------\r\n");

    for (int i = 0; i < size; i++)

    {

        printf("Nhap phan tu thu %d: ", i);

        scanf("%f", &arr[i]);

    }

    printf("---------------\r\n");

}

void xuat(float arr[], int size)

{

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

    for(int i = 0; i < size; i++ )

    {

        printf("%.2f ", arr[i]);

    }

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

}

void swap(float \*a, float \*b)

{

    float temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void giamDan(float arr[], float size)

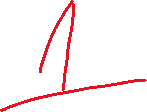
{

    for (int  i = 0; i < size; i++)

    {

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

        {



            if(arr[i] < arr[j])

            {

                swap(&arr[i], &arr[j]);

            }

        }

    }

}

int main()

{

    float a[10] = {0};

    // float a[] = {1.2, 5.6, 9.3, 4.5, 2.3, 6.7, 8.9, 10.5, 16.9, 20.1, 100, 9.3, 9.3, 6.7};

    printf("\nNhap mang float: \r\n");

    nhap(a, SIZE(a));

    printf("Mang chua duoc sap xep\r\n");

    xuat(a, SIZE(a));

    printf("Mang da duoc sap xep giam dan\r\n");

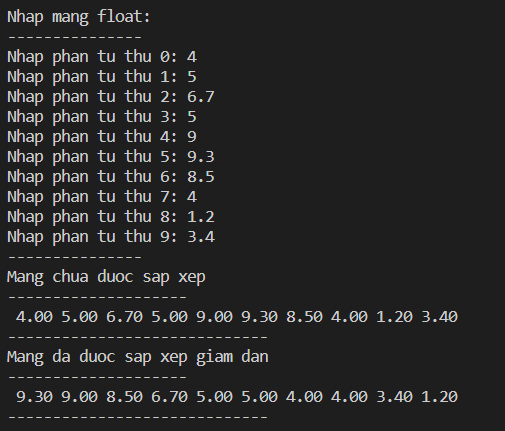
    giamDan(a, SIZE(a));

    xuat(a, SIZE(a));

    printf("\n");

    return 0;

}



1. Tìm số bé thứ n trong câu a và số lớn thứ n trong câu b. Vẽ lưu đồ giải thuật.

#include<stdio.h>

#define SIZE(arr) (sizeof(arr) / sizeof(arr[0]))

void nhap\_int(int arr[], int size)

{

    printf("---------------\r\n");

    for (int i = 0; i < size; i++)

    {

        printf("Nhap phan tu thu %d: ", i);

        scanf("%d", &arr[i]);

    }

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

}

void xuat\_int(int arr[], int size)

{

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

    for(int i = 0; i < size; i++ )

    {

        printf("%d ", arr[i]);

    }

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

}

void nhap\_fl(float arr[], int size)

{

    printf("---------------\r\n");

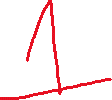
    for (int i = 0; i < size; i++)

    {

        printf("Nhap phan tu thu %d: ", i);

        scanf("%f", &arr[i]);

    }



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

}

void xuat\_fl(float arr[], int size)

{

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

    for(int i = 0; i < size; i++ )

    {

        printf("%.2f ", arr[i]);

    }

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

}

void copy\_int(int arr1[], int arr\_cpy[], int size)

{

    for (int i = 0; i < size; i++)

    {

        arr\_cpy[i] = arr1[i];

    }

}

void copy\_fl(float arr1[], float arr\_cpy[], int size)

{

    for (int i = 0; i < size; i++)

    {

        arr\_cpy[i] = arr1[i];

    }

}

void loc\_mang\_int(int arr[], int size)

{

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

    {

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

        {

            int temp = j;

            if(arr[i] == arr[j]){

                for(int k = 0; k < size - j; k++)

                {

                    arr[temp] = arr[temp + 1];

                    temp++;

                }

                arr[size] = 0;

                size--;

                j--;

            }

        }

    }

}

void loc\_mang\_fl(float arr[], int size)

{

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

    {

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

        {

            int temp = j;

            if(arr[i] == arr[j]){

                for(int k = 0; k < size - j; k++)

                {

                    arr[temp] = arr[temp + 1];

                    temp++;

                }

                arr[size] = 0;

                size--;

                j--;

            }

        }

    }

}

void swap\_int(int \*a, int \*b)

{

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void swap\_fl(float \*a, float \*b)

{

    float temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void tangDan\_int(int arr[], int size)

{

    for (int  i = 0; i < size; i++)

    {

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

        {

            if(arr[i] > arr[j])

            {

                swap\_int(&arr[i], &arr[j]);

            }

        }

    }

}

void giamDan\_fl(float arr[], int size)

{

    for (int  i = 0; i < size; i++)

    {

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

        {

            if(arr[i] < arr[j])

            {

                swap\_fl(&arr[i], &arr[j]);

            }

        }

    }

}

int main()

{

    // int arr\_int[] = {6,6,6,5,9,3,2,7,6,4,10,15,12,2,5,5,9};

    int arr\_int\_cpy[20] = {0};

    int arr\_int[10] = {0};

    printf("Nhap mang int: \r\n");

    nhap\_int(arr\_int,SIZE(arr\_int));

    printf("Mang int vua nhap la: \r\n");

    xuat\_int(arr\_int,SIZE(arr\_int));

    printf("Mang sau khi dc sap sep tang dan la: \r\n");

    tangDan\_int(arr\_int, SIZE(arr\_int));

    xuat\_int(arr\_int,SIZE(arr\_int));

    copy\_int(arr\_int, arr\_int\_cpy, SIZE(arr\_int));

    // printf("Mang sau khi loc: \r\n");

    loc\_mang\_int(arr\_int\_cpy, SIZE(arr\_int));

    // xuat\_int(arr\_int\_cpy, SIZE(arr\_int));

    int n;

    printf("Nhap so thu n: ");

    scanf("%d", &n);

    printf("Phan tu be thu %d trong mang int vua nhap la: %d \r\n\n", n, arr\_int\_cpy[n-1]);

    // float arr\_fl[] = {1.1, 9.6, 5.5, 6.4, 7.3, 2.4, 9.1, 9.1, 5.5, 6.4};

    float arr\_fl\_cpy[20] = {0};

    float arr\_fl[10] = {0};

    printf("Nhap mang float: \r\n");

    nhap\_fl(arr\_fl, SIZE(arr\_fl));

    printf("Mang float vua dc nhap la: \r\n");

    xuat\_fl(arr\_fl, SIZE(arr\_fl));

    printf("Mang sau khi dc sap xep giam dan la: \r\n");

    giamDan\_fl(arr\_fl,SIZE(arr\_fl));

    xuat\_fl(arr\_fl, SIZE(arr\_fl));

    copy\_fl(arr\_fl, arr\_fl\_cpy, SIZE(arr\_fl));

    // printf("mang dc copy la \r\n");

    // xuat\_fl(arr\_fl\_cpy, SIZE(arr\_fl));

    // printf("mang dc loc la: \r\n");

    loc\_mang\_fl(arr\_fl\_cpy,SIZE(arr\_fl));

    // xuat\_fl(arr\_fl\_cpy, SIZE(arr\_fl));

    int n1;

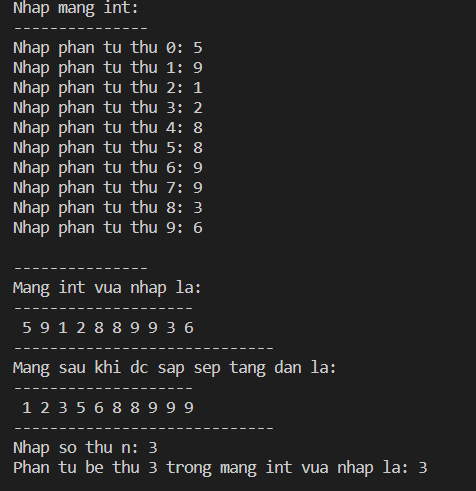
    printf("Nhap so thu n1: ");

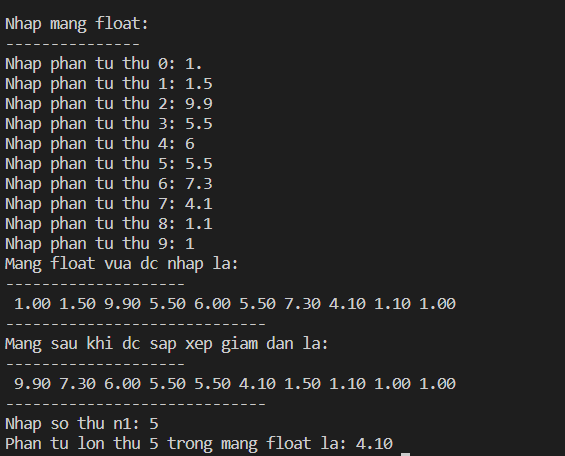
    scanf("%d", &n1);

    printf("Phan tu lon thu %d trong mang float la: %.2f \r\n", n1, arr\_fl\_cpy[n1-1]);

    return 0;

}







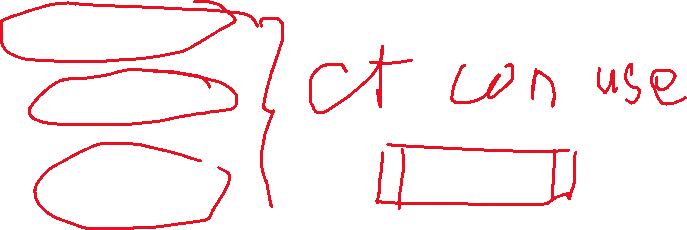
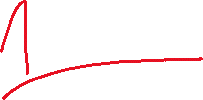
Diagram

Description automatically generated



Diagram

Description automatically generated



1. 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.

#include<stdio.h>

#include<math.h>

#define SIZE(arr) (sizeof(arr) / sizeof(arr[0]))

void nhap\_fl(float arr[], int size)

{

    printf("---------------\r\n");

    for (int i = 0; i < size; i++)

    {

        printf("Nhap phan tu thu %d: ", i);

        scanf("%f", &arr[i]);

    }

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

}

void xuat\_fl(float arr[], int size)

{

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

    for(int i = 0; i < size; i++ )

    {

        printf("%.2f ", arr[i]);

    }

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

}

float tinhTich(float arr[], int size)

{

    float tich = 1;

    for(int i = 0; i < size; i++)

    {

        tich = tich \* arr[i];

    }

    return tich;

}

float tim\_min(float arr[], int size)

{

    float min = arr[0];

    for (int i = 1; i < size; i++)

    {

        if(arr[i] < min){

            min = arr[i];

        }

    }

    return min;

}

void thaypt(float arr[], int size)

{

    for (int i = 0; i < size; i++)

    {

        if(arr[i] < 0) {

            arr[i] = fabs(arr[i]);

        }

    }

}

int main()

{

    // float a[] = {1.2, 5.6, 9.3, 4.5, 2.3, 6.7, 8.9, 10.5, 16.9, 20.1, 100, 9.3, 9.3, 6.7};

    // float a[] = {3.3,4.4,9.9, 5.5, -8.9, 11.1};

    float a[5] = {0};

    printf("\nNhap mang float: \r\n");

    nhap\_fl(a, SIZE(a));

    printf("Mang vua nhap la: \r\n");

    xuat\_fl(a, SIZE(a));

    float tich = tinhTich(a,SIZE(a));

    printf("Tich cac phan tu cua mang la: %.2f \r\n", tich);

    if(tich > 0)

    {

        printf("So nho nhat trong mang la: %.2f \r\n", tim\_min(a, SIZE(a)));

    }

    else if (tich < 0)

    {

        thaypt(a, SIZE(a));

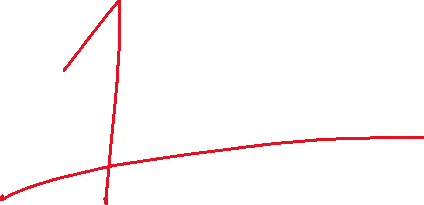
        printf("\nMang sau khi da dc thay la: \r\n");

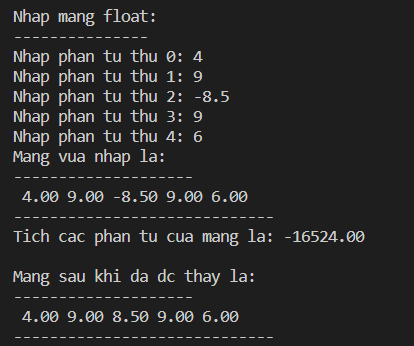
        xuat\_fl(a, SIZE(a));

    }

    return 0;

}





Câu 2 (4đ):

1. Tạo một ma trận NxN và xuất ra màn hình.

#include<stdio.h>

#include<stdlib.h>

void matrix\_init(int matrix[][5], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            matrix[i][j] = rand()%5;

        }

    }

}

void xuat\_matrix(int matrix[][5], int col, int row)

{

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

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            printf(" %2d", matrix[i][j]);

        }

        printf("\r\n");

    }

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

}

int main()

{

    int matrix[5][5] = {0};

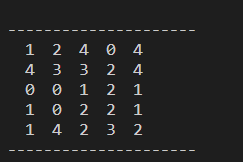
    matrix\_init(matrix, 5, 5);

    xuat\_matrix(matrix, 5, 5);



    return 0;

}





1. Tìm số chẵn đầu tiên trong ma trận, nếu không có thì số lẻ đầu tiên. *Vẽ lưu đồ giải thuật*

#include<stdio.h>

#include<stdlib.h>

void matrix\_init\_rand(int matrix[][5], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            matrix[i][j] = rand()%5;

        }

    }

}

void matrix\_init\_le(int matrix[][5], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            matrix[i][j] = 2 \* (rand() % 10) + 1;

        }

    }

}

void matrix\_init\_chan(int matrix[][5], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            matrix[i][j] = 2 \* (rand() % 10);

        }

    }

}

void xuat\_matrix(int matrix[][5], int col, int row)

{

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

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            printf(" %2d", matrix[i][j]);

        }

        printf("\r\n");

    }

}

int tim\_so\_chan\_dau\_tien(int matrix[][5], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            if(matrix[i][j] % 2 == 0)

            {

                return matrix[i][j];

            }

        }

    }

    return -1;

}

int tim\_so\_le\_dau\_tien(int matrix[][5], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            if(matrix[i][j] % 2 != 0)

            {

                return matrix[i][j];

            }

        }

    }

    return -1;

}

int main()

{

    int matrix[5][5] = {0};

    // matrix\_init\_rand(matrix, 5, 5);

    // matrix\_init\_le(matrix, 5, 5);

    matrix\_init\_chan(matrix, 5, 5);

    // matrix[0][0] = 1;

    xuat\_matrix(matrix, 5, 5);

    int x = tim\_so\_chan\_dau\_tien(matrix,5,5);

    if( x != -1){

        printf("So chan dau tien trong ma tran la : %d\r\n", x);

    }

    else

    {

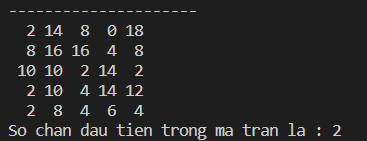
        printf("Khong co so chan trong ma tran \r\n");

        printf("So le dau tien trong ma tran la: %d \r\n", tim\_so\_le\_dau\_tien(matrix,5,5));

    }

    return 0;

}



Diagram

Description automatically generated



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

#include<stdio.h>

void xuat\_matrix(int matrix[][2], int col, int row)

{

    for (int i = 0; i < row; i++)

    {

        for (int j = 0; j < col; j++)

        {

            printf(" %2d", matrix[i][j]);

        }

        printf("\r\n");

    }

}

void nhap(int matrix[2][2])

{

    for (int i = 0; i < 2; i++)

    {

        for(int j = 0; j < 2; j++)

        {

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

        }

    }

}

int Tong(int arr[2][2], int row, int col)

{

    int sum = 0;

    for (int i = 0; i < row; i++)

    {

        for(int j = 0; j < col; j++)

        {

            sum += arr[i][j];

        }

    }

    return sum;

}

int det\_2x2(int arr[2][2])

{

    return arr[0][0] \* arr[1][1] - arr[0][1] \* arr[1][0];

}

int main()

{

    // int maX[2][2] = {4,2,3,9};

    int maX[2][2] = {0};

    printf("\nNhap ma tran 2x2: \r\n");

    nhap(maX);

    printf("ma tran vua nhap la: \r\n");

    xuat\_matrix(maX, 2, 2);

    printf("Tong cac phan tu trong mang la: %d \r\n", Tong(maX,2,2));

    printf("Tinh dinh thuc det = %d \r\n", det\_2x2(maX));

    return 0;

}

