Câu 1 (6đ):

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

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, a[100], tam, i, j;

printf("Nhap so luong so nguyen: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap gia tri

{

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

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

}

for (i=0; i<n; i++) //sap xep gia tri tang dan

{

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

{

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

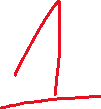
{

tam= a[i];

a[i]= a[j];

a[j]= tam;

}



}

}

printf("Mang da sap xep gia tri tang dan: \n");

for (i=0; i<n; i++) //sau sap xep

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

}

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

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, i, j;

float a[100], tam;

printf("Nhap so luong so thuc: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap gia tri

{

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

scanf("%f", &a[i]);

}

for (i=0; i< n-1; i++) //sap xep gia tri giam dan

{



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

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

{

tam= a[i];

a[i]= a[j];

a[j]= tam;

}

}

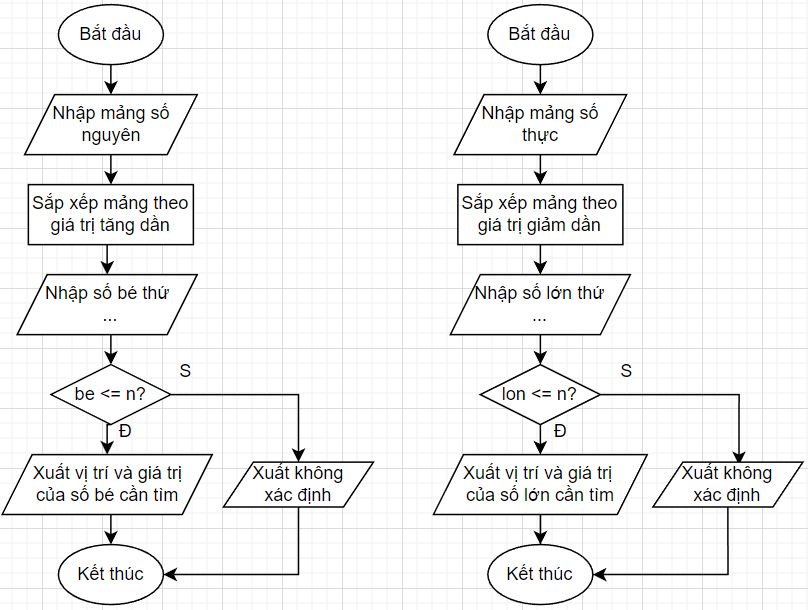
printf("Mang da sap xep gia tri giam dan: \n");

for (i=0; i<n; i++) //sau sap xep

printf("%f \n", a[i]);

}

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





Text

Description automatically generated

#include <stdio.h>

main()

{

int n, a[100], tam, i, j;

printf("Nhap so luong so nguyen: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap gia tri

{

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

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

}

for (i=0; i< n-1; i++) //sap xep gia tri tang dan

{

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

{

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

{

tam= a[i];

a[i]= a[j];

a[j]= tam;

}

}

}

printf("Mang da sap xep gia tri tang dan: \n");

for (i=0; i<n; i++) //sau sap xep

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



//1c //so be thu n

int be;

printf("Tim so be thu: ");

scanf("%d", &be);

if (be<=n)

printf("a[%d] = %d", be-1, a[be-1]);

else

printf("Khong xac dinh");

}

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, i, j;

float a[100], tam;

printf("Nhap so luong so thuc: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap gia tri

{

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

scanf("%f", &a[i]);

}

for (i=0; i< n-1; i++) //sap xep gia tri giam dan

{

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

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

{

tam= a[i];

a[i]= a[j];

a[j]= tam;

}

}

printf("Mang da sap xep gia tri giam dan: \n");

for (i=0; i<n; i++) //sau sap xep

printf("%f \n", a[i]);

//1c //so lon thu n

int lon;

printf("Tim so lon thu: ");

scanf("%d", &lon);

if (lon<=n)

printf("a[%d] = %f \n", lon-1, a[lon-1]);

else

printf("Khong xac dinh \n");

}

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.

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, i, j;

float a[100], tam;

printf("Nhap so luong so thuc: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap gia tri

{

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

scanf("%f", &a[i]);

}

for (i=0; i< n-1; i++) //sap xep gia tri giam dan

{

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

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

{

tam= a[i];

a[i]= a[j];

a[j]= tam;

}

}

printf("Mang da sap xep gia tri giam dan: \n");

for (i=0; i<n; i++) //sau sap xep

printf("%f \n", a[i]);

// //1c //so lon thu n

// int lon;

// printf("Tim so lon thu: ");

// scanf("%d", &lon);

// if (lon<=n)

// printf("a[%d] = %f \n", lon-1, a[lon-1]);

// else

// printf("Khong xac dinh \n");

//1d //tich cua mang

float tich=1;

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

tich\*=a[i];

printf("Tich cua mang: %f \n", tich);

if (tich<0) //so am? -> pt am thanh duong

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

if (a[i]<0)



a[i]/=-1;



else if (tich>0) //so duong? =min(mang)

tich=a[n-1];



printf("Mang sau khi dieu chinh theo tich: \n");

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

printf("%f \n", a[i]);

}

Câu 2 (4đ):

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

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, a[10][10], i, j;

printf("Kich thuoc ma tran vuong: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap



{

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

{

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

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

}

}

printf("Mang: \n");

for (i=0; i<n; i++) //xuat

{

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

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

printf("\n");

}

}

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

Diagram

Description automatically generated

Diagram

Description automatically generated

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, a[10][10], i, j;

printf("Kich thuoc ma tran vuong: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap

{

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

{

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

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

}

}

printf("Mang: \n");

for (i=0; i<n; i++) //xuat

{

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

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

printf("\n");

}

//b //tim so chan dau tien, khong co thi so le dau tien

int chan=0;

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

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

{



if (chan==1)

break;

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

{

printf("So chan dau tien: a[%d][%d] = %d \n", i, j, a[i][j]);

chan=1;

}

}

//ma tran 0 chan thi so dau tien la le

if (chan==0)

printf("So le dau tien: a[0][0] = %d \n", a[0][0]);

}

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

Text

Description automatically generated

#include <stdio.h>

main()

{

int n, a[10][10], i, j;

printf("Kich thuoc ma tran vuong: ");

scanf("%d", &n);

for (i=0; i<n; i++) //nhap

{

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

{

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

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

}

}

printf("Mang: \n");

for (i=0; i<n; i++) //xuat

{

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

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

printf("\n");

}

//c //tong, dinh thuc cua ma tran

int tong=0; //tong

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

{

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

tong+=a[i][j];

}

printf("Tong cua ma tran: %d \n", tong);

int dt=0; //dinh thuc



if(n==1)

dt= a[0][0];

else if(n==2) //ma tran NxN lon hon 2 chua bt lam nen dat tam det(a[2][2])

dt= (a[0][0] \*a[1][1] - a[0][1] \*a[1][0]);

printf("Dinh thuc cua ma tran bac %d: %d", n, dt);

//ma tran NxN bac lon hon 2 chua bt lam

}