**Họ và tên: Trần Thẩm Hoàng Long**

**MSSV: 21138301**

**BÀI TẬP BUỔI 2 C**

1. (6đ)
2. Nhập vào dãy số nguyên và sắp xếp giá trị tăng dần.

|  |  |
| --- | --- |
| ***Code:***  #include <stdio.h>  #define SIZE 20  void NhapMang(int a[], int &n){  do{  printf ("nhap so luong phan tu n: ");  scanf ("%d", &n);  } while(n < 1 || n > SIZE);  for (int i = 0; i < n; i++){  printf ("Nhap phan tu thu %d: ", i);  scanf("%d", &a[i]);  }  }  void XuatMang(int a[], int n){  printf ("Noi dung cua mang la: ");  for (int i = 0; i < n; i++){  printf ("%d ", a[i]);  }  printf ("\n");  }  void SapXepTangDan(int a[], int n){  int temp;  for(int i = 0; i < n - 1; i++){  for(int j = i + 1; j < n; j++){  if(a[i] > a[j]){  // Hoan vi 2 so a[i] va a[j]  temp = a[i];  a[i] = a[j];  a[j] = temp;  }  }  }  }  int main(){  int a[SIZE], n;  NhapMang(a, n);  XuatMang(a, n);  printf("\nDay so sau khi sap xep tang dan: \n");  SapXepTangDan(a, n);  XuatMang(a, n);  } | ***Result:*** |

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

|  |  |
| --- | --- |
| ***Code:***  #include <stdio.h>  #define SIZE 20  void NhapMang(float a[], int &n){  do{  printf ("nhap so luong phan tu n: ");  scanf ("%d", &n);  } while(n < 1 || n > SIZE);  for (int i = 0; i < n; i++){  printf ("Nhap phan tu thu %d: ", i);  scanf("%f", &a[i]);  }  }  void XuatMang(float a[], int n){  printf ("Noi dung cua mang la: ");  for (int i = 0; i < n; i++){  printf ("%.1f ", a[i]);  }  printf ("\n");  }  void SapXepGiamDan(float a[], int n){  float temp;  for(int i = 0; i < n - 1; i++){  for(int j = i + 1; j < n; j++){  if(a[i] < a[j]){  // Hoan vi 2 so a[i] va a[j]  temp = a[i];  a[i] = a[j];  a[j] = temp;  }  }  }  }  int main(){  float a[SIZE];  int n;  NhapMang(a, n);  XuatMang(a, n);  printf("\nDay so sau khi sap xep giam dan: \n");  SapXepGiamDan(a, n);  XuatMang(a, n);  } | ***Result:*** |

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

* ***Flowchart:***

Diagram

Description automatically generated

* ***Code:***

#include <stdio.h>

#define SIZE 20

void NhapMang(int a[], int &n){

do{

printf ("nhap so luong phan tu n: ");

scanf ("%d", &n);

} while(n < 1 || n > SIZE);

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

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

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

}

}

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

printf ("Noi dung cua mang la: ");

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

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

}

printf ("\n");

}

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

int temp;

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

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

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

// Hoan vi 2 so a[i] va a[j]

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}



}

int main(){

int a[SIZE], n, k;

NhapMang(a, k);

XuatMang(a, k);

printf("\nDay so sau khi sap xep tang dan: \n");

SapXepTangDan(a, k);

XuatMang(a, k);

printf("Nhap n: ");

scanf("%d", &n);

printf("So be thu %d trong mang tren la: %d", n, a[n - 1]);

}

* ***Result:***

Text

Description automatically generated

1. Tìm số lớn thứ n:

* *Flowchart:*

Diagram

Description automatically generated

* ***Code:***

#include <stdio.h>

#define SIZE 20

void NhapMang(float a[], int &n){

do{

printf ("nhap so luong phan tu n: ");

scanf ("%d", &n);

} while(n < 1 || n > SIZE);

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

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

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

}

}

void XuatMang(float a[], int n){

printf ("Noi dung cua mang la: ");

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

printf ("%.1f ", a[i]);

}

printf ("\n");

}

void SapXepGiamDan(float a[], int n){

float temp;

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

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

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

// Hoan vi 2 so a[i] va a[j]

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

int main(){



float a[SIZE];

int k, n;

NhapMang(a, k);

XuatMang(a, k);

printf("\nDay so sau khi sap xep giam dan: \n");

SapXepGiamDan(a, k);

XuatMang(a, k);

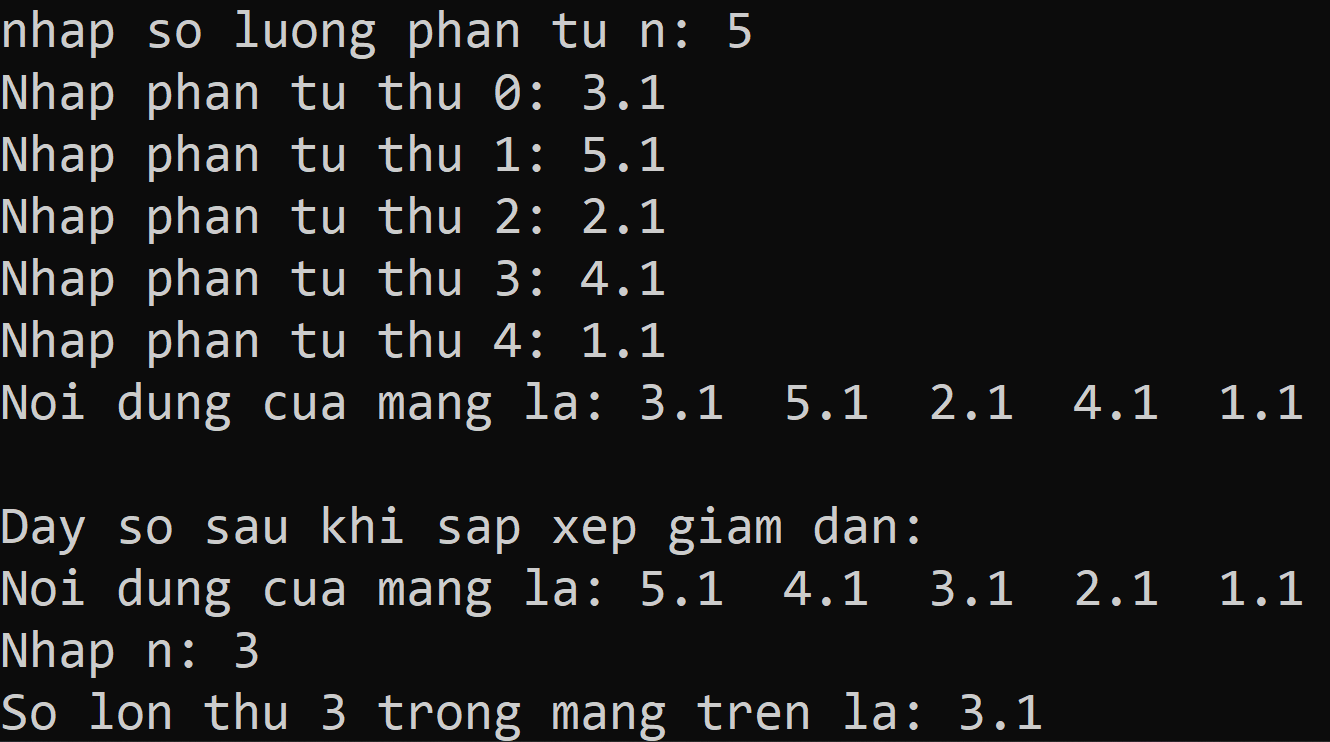
printf("Nhap n: ");

scanf("%d", &n);

printf("So lon thu %d trong mang tren la: %.1f", n, a[n - 1]);

}

* ***Result:***



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.
* ***Code:***

#include <stdio.h>

#define SIZE 20

void NhapMang(float a[], int &n){

do{

printf ("nhap so luong phan tu n: ");

scanf ("%d", &n);

} while(n < 1 || n > SIZE);

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

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

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

}

}

void XuatMang(float a[], int n){

printf ("\nNoi dung cua mang la: ");

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

printf ("%.1f ", a[i]);

}

printf ("\n");

}

float tichcuaMang(float a[], int n){

float ketqua = 1.0;

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

ketqua = ketqua \* a[i];

}

return ketqua;

}

float min(float a[], int n){

float min = a[0];

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

if(a[i] < min){

min = a[i];

}

}

return min;

}

void ChuyenAmtoDuong(float a[], int n){

printf ("\nChuyen so am thanh duong la: ");

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

if(a[i] < 0){

a[i] = -a[i];

}

printf ("%.1f ", a[i]);

}

printf ("\n");

}

int main(){

float a[SIZE];

int n;

NhapMang(a, n);

XuatMang(a, n);

printf("\nTich cua mang tren la: %.1f", tichcuaMang(a, n));

if(tichcuaMang(a, n) > 0)

printf("\nSo be nhat trong mang la: %.1f", min(a, n));

else



ChuyenAmtoDuong(a, n);

}

* ***Result:***

|  |  |
| --- | --- |
| **TH1: Tích là số âm** | **TH2: Tích là số dương** |

1. (4đ)

a. Tạo một ma trận N x N và xuất ra màn hình

|  |  |
| --- | --- |
| ***Code:***  #include <stdio.h>  #define SIZE1 100  #define SIZE2 100  void NhapMaTran(int a[][SIZE1], int &m, int &n){  printf("Nhap so dong cua ma tran: ");  scanf("%d", &m);  printf("Nhap so cot cua ma tran: ");  scanf("%d", &n);  for(int i = 0; i < m; i++){  for (int j = 0; j < n; j++){  printf("Nhap a[%d][%d]: ", i, j);  scanf("%d", &a[i][j]);  }  }  }  void XuatMaTran(int a[][SIZE1], int m, int n){  for(int i = 0; i < m; i++){  for(int j = 0; j < n; j++){  printf("%d ", a[i][j]);  }  printf("\n");  }  }  int main(){  int a[SIZE1][SIZE2], m, n;  NhapMaTran(a, m, n);  XuatMaTran(a, m, n);  } | ***Result:*** |

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

- ***Flowchart:***



Diagram

Description automatically generated

* ***Code:***

#include <stdio.h>

#define SIZE1 100

#define SIZE2 100

void NhapMaTran(int a[][SIZE1], int &m, int &n){

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

scanf("%d", &m);

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

scanf("%d", &n);

for(int i = 0; i < m; i++){

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

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

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



}

}

}

void XuatMaTran(int a[][SIZE1], int m, int n){

for(int i = 0; i < m; i++){

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

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

}

printf("\n");

}

}

int soChandautien(int a[][SIZE1], int m, int n){

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

return a[0][0];

return -1;

}

int soLedautien(int a[][SIZE1], int m, int n){

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

return a[0][0];

return -1;

}

int main(){

int a[SIZE1][SIZE2], m, n;

NhapMaTran(a, m, n);

XuatMaTran(a, m, n);

if(soChandautien(a, m, n) == -1)

printf("So le dau tien cua ma tran la: %d", soLedautien(a, m, n));

else if(soLedautien(a, m, n) == -1)

printf("So chan dau tien cua ma tran la: %d", soChandautien(a, m, n));

return 0;

}

* ***Result:***

|  |  |
| --- | --- |
|  |  |

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

***- Code:***

#include <stdio.h>

#include <math.h>

#define SIZE1 100

#define SIZE2 100

void NhapMaTran(int a[][SIZE1], int &m, int &n){

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

scanf("%d", &m);

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

scanf("%d", &n);

for(int i = 0; i < m; i++){

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

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

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

}

}

}

void XuatMaTran(int a[][SIZE1], int m, int n){

for(int i = 0; i < m; i++){

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

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

}

printf("\n");

}

}

int TongMaTran(int a[][SIZE1], int m, int n){

int sum = 0;

for(int i = 0; i < m; i++) {

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



sum = sum + a[i][j];

}

}

return sum;

}

int DinhThuc2x2(int a[][SIZE1], int n){



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



return det;

}

int main(){

int a[SIZE1][SIZE2], m, n;

NhapMaTran(a, m, n);

XuatMaTran(a, m, n);

printf("\nDinh thuc cua ma tran tren la: %d", DinhThuc2x2(a, n));

printf("\nTong cac phan tu trong ma tran tren la: %d", TongMaTran(a, m, n));

}

* ***Result:***

Text

Description automatically generated

