Proiect

Bazele Programării Calculatoarelor

Vectori:

1. Sa se scrie programul care realizeaza cautarea unei valori date intr-un vector

#include <stdio.h>

//Cautarea unei valori date intr-un vector

int main() {

int n;

printf ("Introduceti dimensiunea vectorului, m=");

scanf ("%d",&n);

int v[100];

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

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

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

}

int value;

printf("Introduceti valoarea cautata");

scanf ("%d", &value);

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

if (v[i]==value){

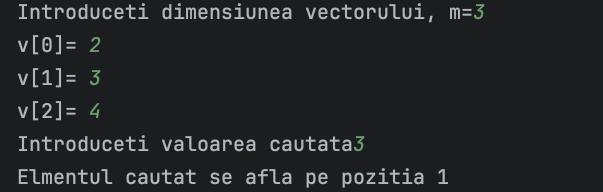
printf ("Elmentul cautat se afla pe pozitia %d", i);

}

}

return 0;

}



1. Sa se scrie programul care realizeaza suma elementelor dintr-un vector de numere intregi.

#include <stdio.h>

int main() {

int n; // Numarul de elemente ale vectorului

printf ("Introduceti dimensiunea vectorului, m=");

scanf ("%d",&n);

int v[100];

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

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

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

}

int sum = 0;

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

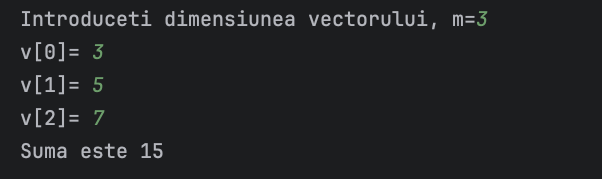
sum +=v[i];

}

printf ("Suma este %d \n ", sum);

return 0;

}



1. Determinarea elementului maxim intr-un vector de numere intregi.

#include <stdio.h>

int main() {

int n;

printf ("Introduceti dimensiunea vectorului, m=");

scanf ("%d",&n);

int v[100];

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

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

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

}

int max;

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

if(max < v[i])

{

max=v[i];

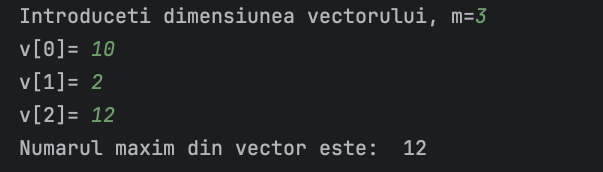
}

}

printf ("Numarul maxim din vector este: %d \n ", max);

return 0;

}



1. Sa se scrie programul C care determina produsul scalar a doi vectori de dimensiune n

#include <stdio.h>

int main() {

int n;

printf ("Introduceti dimensiunea vectorului, m=");

scanf ("%d",&n);

int v[100];

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

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

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

}

int m;

printf ("Introduceti dimensiunea vectorului, m=");

scanf ("%d",&m);

int y[100];

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

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

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

}

int produsScalar=0;

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

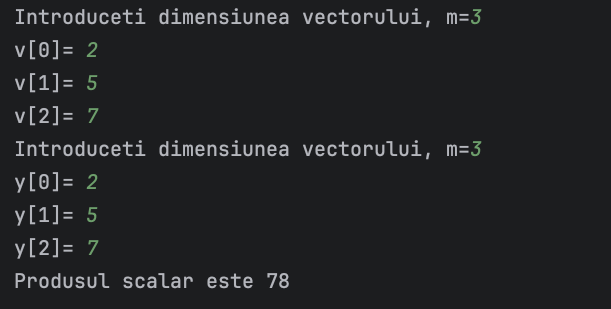
produsScalar+=v[i]\*y[i];

}

printf ("Produsul scalar este %d \n", produsScalar);

return 0;

}



1. Sa se realizeze sortarea prin selectie pentru un vector de numere intregi.

#include <stdio.h>

int main() {

int n;

printf("Introduceti numarul de elemente in vector: ");

scanf("%d", &n);

int arr[100];

printf("Introduceti elementele vectorului:\n");

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

printf("Elementul %d: ", i + 1);

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

}

printf("Vectorul initial:\n");

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

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

}

printf("\n");

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

int min\_idx = i;

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

if (arr[j] < arr[min\_idx]) {

min\_idx = j;

}

}

int temp = arr[min\_idx];

arr[min\_idx] = arr[i];

arr[i] = temp;

}

printf("Vectorul sortat:\n");

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

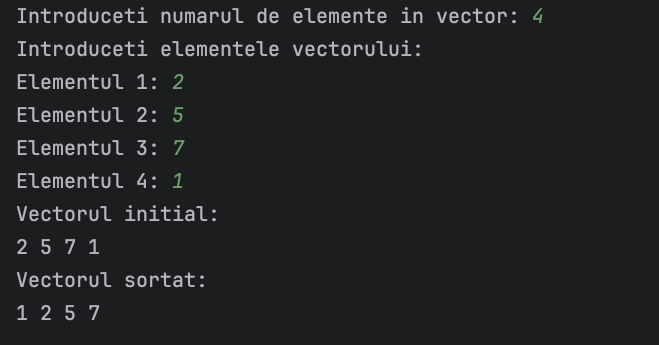
printf("%d ", arr[i]);

}

printf("\n");

return 0;

}



1. Sa se scrie programul C care realizeaza inserarea unui element dat intr-un vector ordonat de numere intregi. dupa inserare vectorul va ramane ordonat.

#include <stdio.h>

int main() {

int n;

float x[100], a;

printf("Dimensiunea vectorului: ");

scanf("%d", &n);

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

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

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

}

printf("Introduceti valoarea de inserat in vector: ");

scanf("%f", &a);

int i = 0;

while ((i < n) && (x[i] < a)) {

i++;

}

for (int j = n; j > i; j--) {

x[j] = x[j - 1];

}

x[i] = a;

n++;

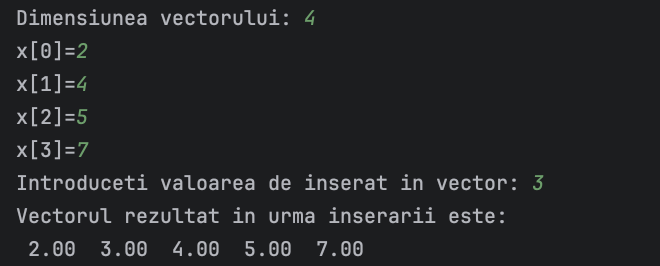
printf("Vectorul rezultat in urma inserarii este:\n");

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

printf("%5.2f ", x[i]);

}

}



1. Sa se scrie programul C care realizeaza interclasarea (merge-ul) a doi vectori de numere intregi.

#include <stdio.h>

int main() {

int n1, n2;

printf("Introduceti dimensiunea primului vector: ");

scanf("%d", &n1);

int arr1[100];

printf("Introduceti elementele primului vector:\n");

for (int i = 0; i < n1; i++) {

printf("Elementul %d: ", i + 1);

scanf("%d", &arr1[i]);

}

printf("Introduceti dimensiunea celui de-al doilea vector: ");

scanf("%d", &n2);

int arr2[100];

printf("Introduceti elementele celui de-al doilea vector:\n");

for (int i = 0; i < n2; i++) {

printf("Elementul %d: ", i + 1);

scanf("%d", &arr2[i]);

}

int result[100];

int i = 0, j = 0, k = 0;

while (i < n1 && j < n2) {

if (arr1[i] < arr2[j]) {

result[k++] = arr1[i++];

}

else {

result[k++] = arr2[j++];

}

}

while (i < n1) {

result[k++] = arr1[i++];

}

while (j < n2) {

result[k++] = arr2[j++];

}

printf("Vectorul rezultat dupa merge este: ");

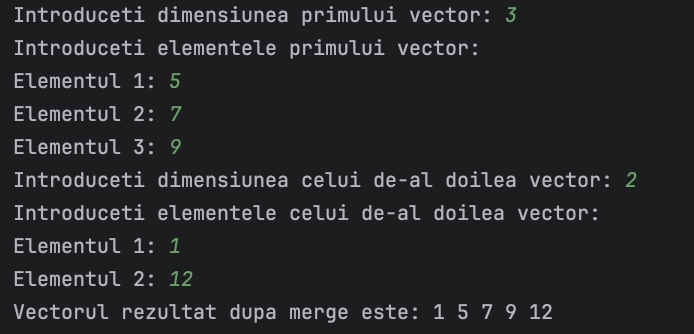
for (i = 0; i < n1 + n2; i++) {

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

}

return 0;

}



1. Sa se scrie programul C care realizeaza produsul vectorial a doi vectori de numere intregi.

#include <stdio.h>

int main() {

int n;

printf("Introduceti dimensiunea vectorilor: ");

scanf("%d", &n);

int arr1[100], arr2[100], result[100];

printf("Introduceti elementele primului vector:\n");

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

printf("Elementul %d: ", i + 1);

scanf("%d", &arr1[i]);

}

printf("Introduceti elementele celui de-al doilea vector:\n");

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

printf("Elementul %d: ", i + 1);

scanf("%d", &arr2[i]);

}

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

result[i] = arr1[i] \* arr2[i];

}

printf("Produsul vectorial este: ");

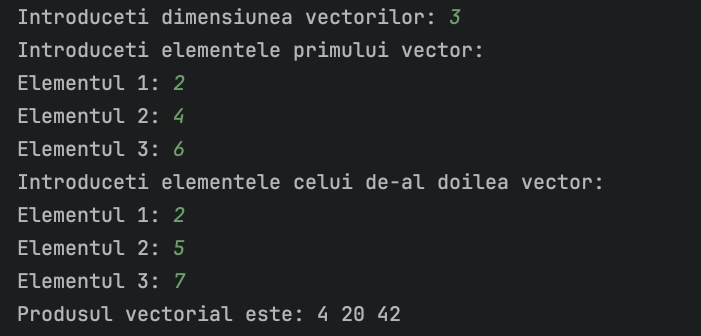
for (int i = 0; i < n; i++) {

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

}

return 0;

}



1. Sa se scrie programul C care determina minimul, precum si prima, ultima si penultima apritie a minimului intr-un vector de numere intregi.

#include <stdio.h>

int main() {

int n;

printf("Introduceti dimensiunea vectorului: ");

scanf("%d", &n);

int v[100];

printf("Introduceti elementele vectorului:\n");

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

printf("Elementul %d: ", i + 1);

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

}

int min = v[0];

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

if (v[i] < min) {

min = v[i];

}

}

int primaAp = -1, ultimaAp = -1, penultimaAp = -1;

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

if (v[i] == min) {

if (primaAp == -1) {

primaAp = i;

}

penultimaAp = ultimaAp;

ultimaAp = i;

}

}

printf("Minimul este: %d\n", min);

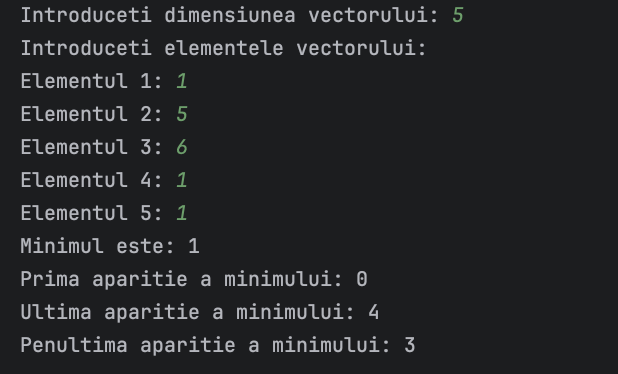
printf("Prima aparitie a minimului: %d\n", primaAp);

printf("Ultima aparitie a minimului: %d\n", ultimaAp);

printf("Penultima aparitie a minimului: %d\n", penultimaAp);

return 0;

}



1. Sa se scrie programul C care realizeaza diferenta dintre un vector v2 si un vector1. Vectorul rezultat va contine elementele care se regasesc in v2 dar care nu se regasesc in v1.

#include <stdio.h>

int main() {

int n;

printf("Introduceti dimensiunea vectorilor: ");

scanf("%d", &n);

int v1[100], v2[100], result[100];

printf("Introduceti elementele primului vector:\n");

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

printf("Elementul %d: ", i + 1);

scanf("%d", &v1[i]);

}

printf("Introduceti elementele celui de-al doilea vector:\n");

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

printf("Elementul %d: ", i + 1);

scanf("%d", &v2[i]);

}

int count = 0;

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

int found = 0;

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

if (v1[i] == v2[j]) {

found = 1;

break;

}

}

if (!found) {

result[count++] = v2[i];

}

}

printf("Vectorul rezultat este: ");

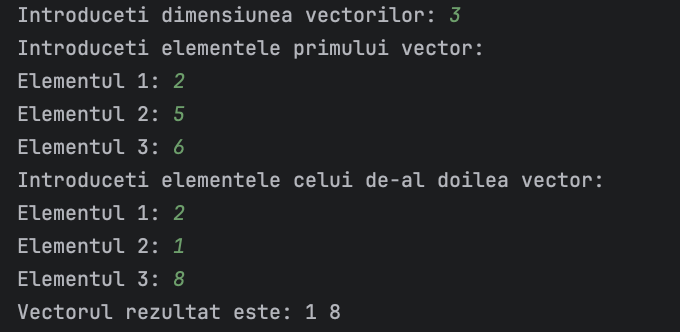
for (int i = 0; i < count; i++) {

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

}

return 0;

}



Matrice:

1. Sa se realizeze programul care determina elementele maxim si minim dintr-o matrice dreptungiulara.

#include <stdio.h>

int main() {

int m, n;

int a[100][100];

printf("Numar de linii: ");

scanf("%d", &m);

printf("Numar de coloane: ");

scanf("%d", &n);

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

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

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

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

}

}

printf("\nMatricea introdusa:\n");

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

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

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

}

printf("\n");

}

int min = a[0][0];

int max = a[0][0];

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

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

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

max = a[i][j];

}

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

min = a[i][j];

}

}

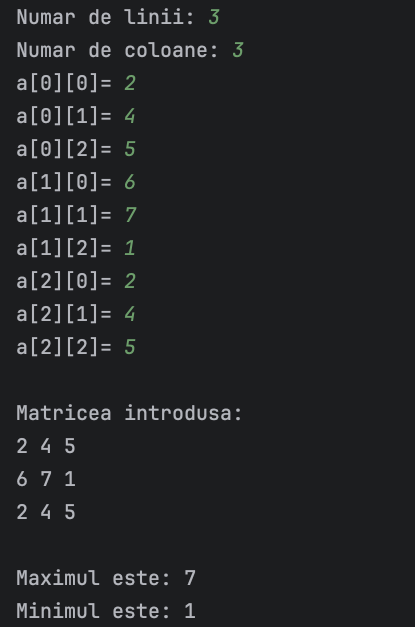
}

printf("\nMaximul este: %d", max);

printf("\nMinimul este: %d\n", min);

return 0;

}



1. Sa se creeze un program care realizeaza inmultirea a doua matrice dreptunghiulare

#include <stdio.h>

int main() {

int a[100][100], b[100][100], c[100][100];

int m, n, p, k;

printf("Matricea 1:\n");

printf("Numarul de linii: ");

scanf("%d", &m);

printf("Numarul de coloane: ");

scanf("%d", &n);

printf("Introduceti elementele matricei 1:\n");

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

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

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

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

}

}

printf("Matricea 1:\n");

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

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

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

}

printf("\n");

}

printf("Matricea 2:\n");

printf("Numarul de coloane: ");

scanf("%d", &p);

printf("Introduceti elementele matricei 2:\n");

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

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

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

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

}

}

printf("Matricea 2:\n");

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

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

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

}

printf("\n");

}

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

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

c[i][j] = 0;

for (k = 0; k < n; k++) {

c[i][j] = c[i][j] + a[i][k] \* b[k][j];

}

}

}

printf("Matricea rezultat este:\n");

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

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

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

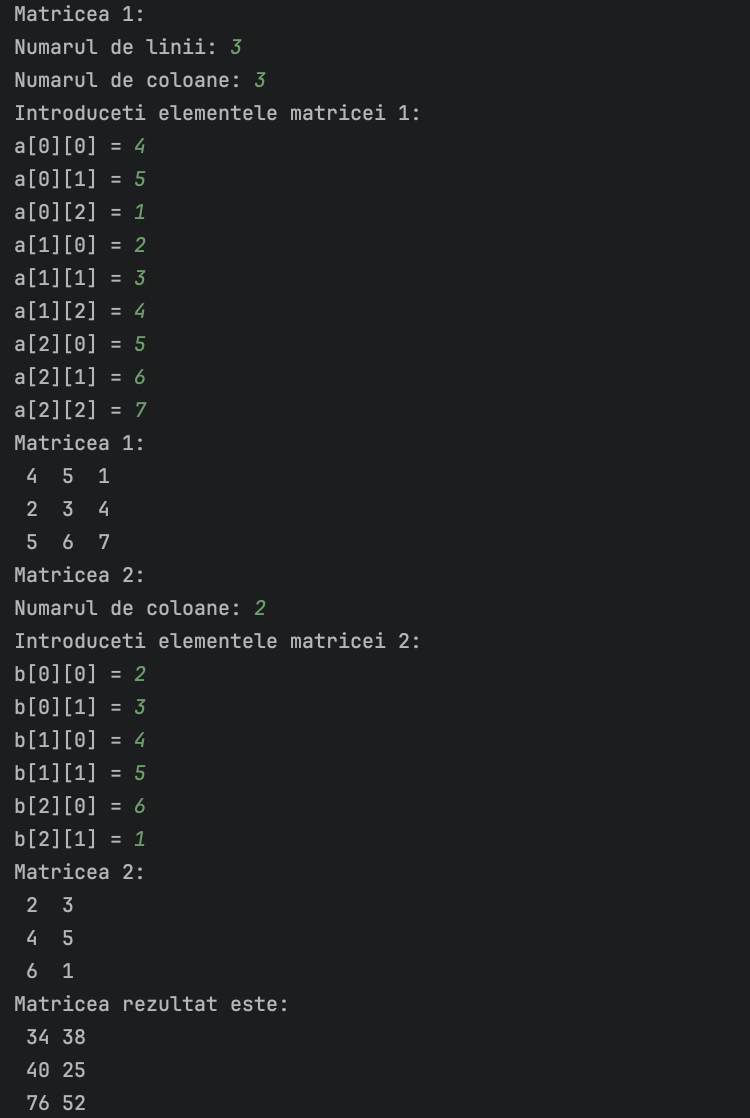
}

printf("\n");

}

return 0;

}



1. Sa se scrie programul care ridica matricea la o putere (valoarea puterii este citita de la tastatura)

#include <stdio.h>

int main() {

int m, p, i, j, k;

int a[10][10], b[10][10], c[10][10];

printf("\nMatricea 1\n");

printf("Nr de linii/coloane = ");

scanf("%d", &m);

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

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

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

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

}

}

printf("Matricea a este:\n");

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

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

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

}

printf("\n");

}

printf("Puterea este: ");

scanf("%d", &p);

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

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

b[i][j] = (i == j) ? 1 : 0;

}

}

for (int q = 0; q < p; q++) {

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

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

c[i][j] = 0;

for (k = 0; k < m; k++) {

c[i][j] += b[i][k] \* a[k][j];

}

}

}

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

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

b[i][j] = c[i][j];

}

}

}

printf("Matricea rezultat este:\n");

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

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

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

}

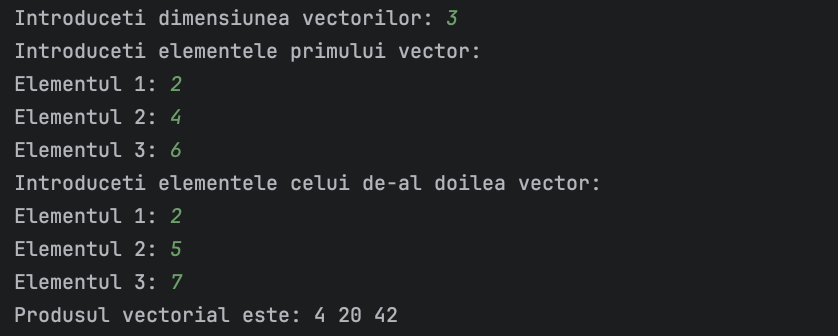
printf("\n");

}

getchar();

return 0;

}



1. Sa se scrie programul C care sorteaza crescator elementele de pe diagonala principala, folosind interschimbarea liniilor si coloanelor

#include <stdio.h>

int main() {

int i, j, n, p, aux;

printf("Numarul de linii/coloane = ");

scanf("%d", &n);

int a[10][10];

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

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

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

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

}

}

printf("Matrice introdusa:\n");

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

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

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

}

printf("\n");

}

do {

p = 0;

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

if (a[i][i] > a[i + 1][i + 1]) {

int l = i;

int ll = i + 1;

p = 1;

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

aux = a[l][j];

a[l][j] = a[ll][j];

a[ll][j] = aux;

}

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

aux = a[j][l];

a[j][l] = a[j][ll];

a[j][ll] = aux;

}

}

}

} while (p);

printf("\nMatrice rezultat:\n");

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

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

printf("%4d ", a[i][j]);

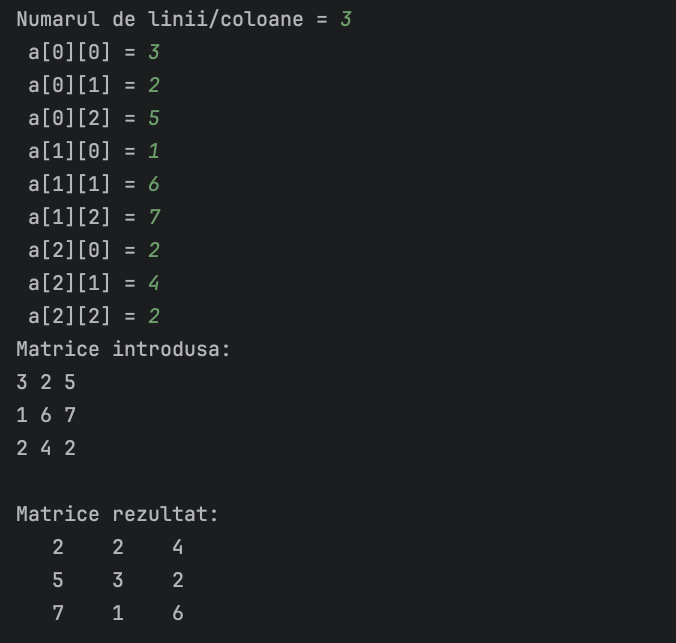
}

printf("\n");

}

return 0;

}



1. Sa se scrie programul C prin care sa se afiseze o matrice sub forma de "scara" folosind caracterul "#". Elementele se vor insera pe diagonala secundara

#include <stdio.h>

int main() {

int n;

printf("Introduceti numarul de linii/coloane: ");

scanf("%d", &n);

char stairCase[100][100];

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

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

if (i + j >= n - 1) {

stairCase[i][j] = '#';

}

else {

stairCase[i][j] = ' ';

}

}

}

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

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

printf("%c", stairCase[i][j]);

}

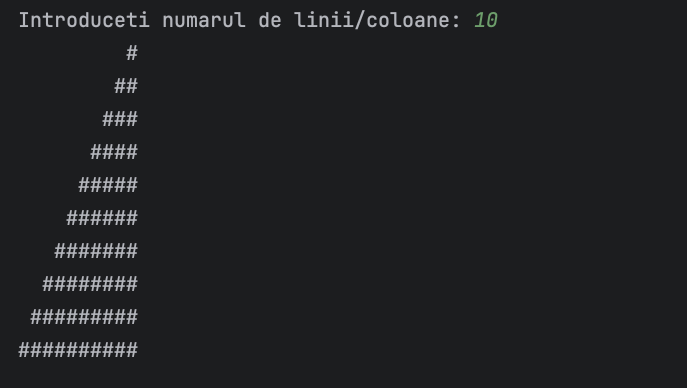
printf("\n");

}

printf("\n");

return 0;

}



1. Sa se realizeze programul care determina produsul dintre un vector si o matrice

#include <stdio.h>

int main() {

int n, p, j, i, k;

int x[10], a[10][10], y[10];

printf("\nNr de elemente ale vectorului= ");

scanf("%d", &n);

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

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

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

}

printf("\nMatricea ");

printf("\nNr de coloane= ");

scanf("%d", &p);

// Citirea matricei a

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

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

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

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

}

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

y[j] = 0;

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

y[j] += x[k] \* a[k][j];

}

printf("\nRezultatul este: \n");

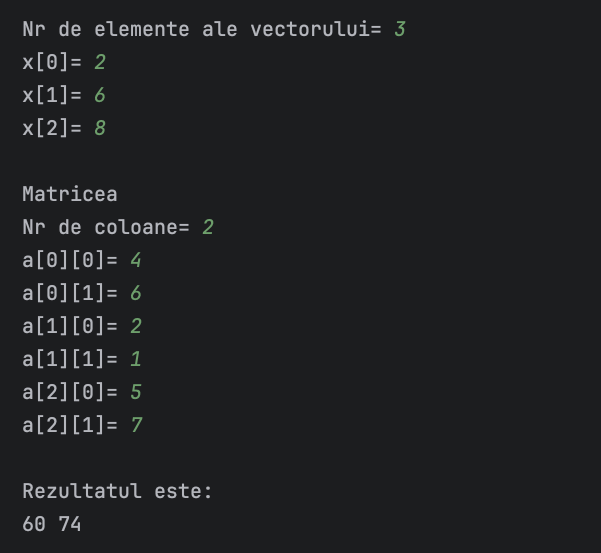
for (j = 0; j < p; j++)

printf("%d ", y[j]);

printf("\n");

return 0;

}



1. Sa se determine elementul maxim din triunghiul de deasupra diagonelelor unei matrice patratice(exclusiv diagonalele)

#include <stdio.h>

int main() {

float a[10][10], max;

int m, i, j;

printf("m = ");

scanf("%d", &m);

// Citirea matricei

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

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

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

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

}

}

// Inițializarea lui max cu primul element din triunghi

max = a[0][1];

// Găsirea maximului din triunghiul de deasupra diagonalelor

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

for (j = i + 1; j < m - i; j++) {

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

max = a[i][j];

}

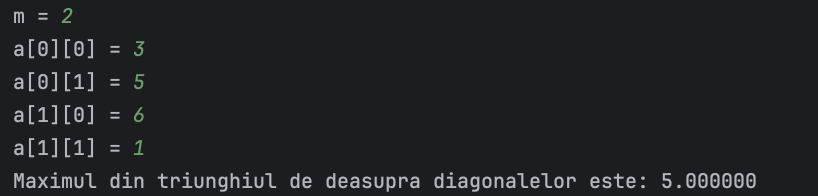
}

}

printf("Maximul din triunghiul de deasupra diagonalelor este: %f", max);

return 0;

}



1. Sa se sorteze prima linie a unei matrice fara a schimba structura coloanelor

#include <stdio.h>

int main() {

float a[10][10], aux;

int n, m, i, j, k;

printf("m = ");

scanf("%d", &m);

printf("n = ");

scanf("%d", &n);

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

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

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

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

}

}

k = 1;

while (k) {

k = 0;

for (j = 0; j < m - 1; j++) {

if (a[0][j] > a[0][j + 1]) {

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

aux = a[i][j];

a[i][j] = a[i][j + 1];

a[i][j + 1] = aux;

k = 1;

}

}

}

}

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

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

printf("%5.2f", a[i][j]);

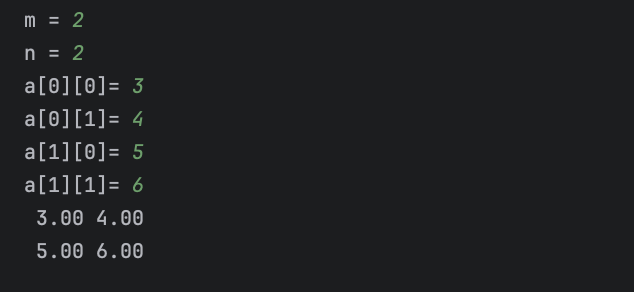
}

printf("\n");

}

return 0;

}



1. Pentru fiecare linie a unei matrice sa se scada din elementele sale valoarea minima din acea linie

#include <stdio.h>

int main() {

float a[10][10], min;

int n, m, i, j;

printf("m=");

scanf("%d", &m);

printf("n=");

scanf("%d", &n);

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

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

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

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

}

}

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

min = a[i][0];

for (j = 1; j < m; j++) {

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

min = a[i][j];

}

}

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

a[i][j] = a[i][j] - min;

}

}

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

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

printf("%3.2f ", a[i][j]);

}

printf("\n");

}

return 0;

}



1. Sa se realizeze programul care calculeaza media aritmetica elementelor de pe fiecare linie si coloana a unei matrice dreptunghiulare avand elementele citite de la tastatura.

#include <stdio.h>

int main() {

int n, m, a[100][100];

printf("Numarul de linii: ");

scanf("%d", &m);

printf("Numarul de coloane: ");

scanf("%d", &n);

if (m > 100 || m <= 0 || n > 100 || n <= 0) {

printf("Dimensiunile matricei sunt invalide.\n");

return 1;

}

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

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

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

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

}

}

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

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

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

}

printf("\n");

}

printf("\nMedia pe fiecare linie:\n");

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

int suma = 0;

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

suma += a[i][j];

}

float avg = (float)suma / n;

printf("Linia %d: %.2f\n", i + 1, avg);

}

printf("\nMedia pe fiecare coloana:\n");

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

int suma = 0;

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

suma += a[i][j];

}

float avg = (float)suma / m;

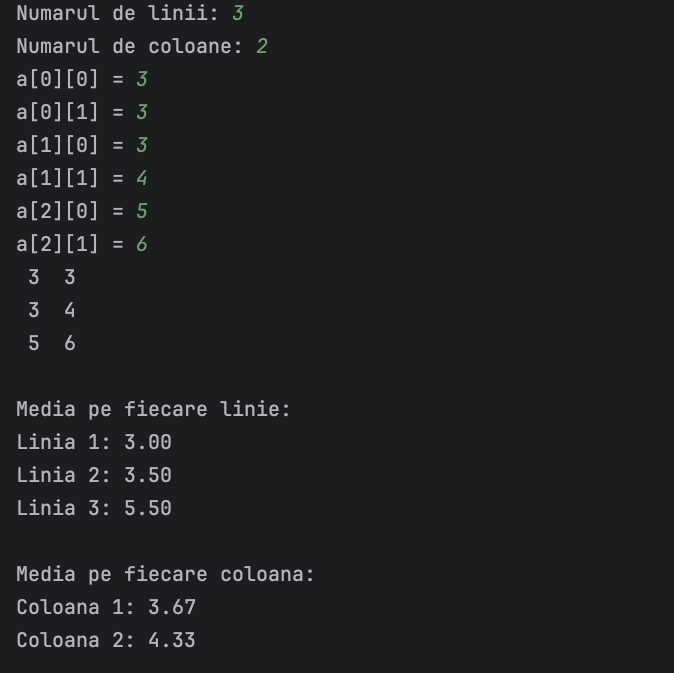
printf("Coloana %d: %.2f\n", j + 1, avg);

}

getchar();

return 0;

}



Subprograme:

#include <stdio.h>

int citire(float x[]) {

int i, n;

printf("n=");

scanf("%d", &n);

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

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

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

}

return n;

}

// Subprogram pentru citirea elementelor intregi dintr-un vector cu returnarea numarului de elemente prin numele subprogramului

int citire2(int x[]) {

int i, n;

printf("n=");

scanf("%d", &n);

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

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

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

}

return n;

}

// Subprogram pentru afisarea elementelor din vector de numere reale

void afisare(float x[], int n) {

int i;

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

printf("%5.2f ", x[i]);

}

}

// Subprogram pentru afisarea elementelor din vector de numere intregi

void afisare2(int x[], int n) {

int i;

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

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

}

}

// Subprogram care determina valoarea medie a vanzarilor unei firme, eliminand zilele in care nu s-au inregistrat vanzari

float medieVanzariZilnice(float x[], int n) {

int i, k = 0;

float medie = 0;

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

if (x[i] != 0) {

medie += x[i];

k++;

}

}

if (k) {

medie /= k;

}

return medie;

}

// Subprogram care determina numarul de aparitii ale unei valori intregi intr-un vector

int numarAparitii(int vector[], int lungime, int target) {

int aparitii = 0;

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

if (vector[i] == target) {

aparitii++;

}

}

return aparitii;

}

// Subprogram pentru compactarea unui vector prin eliminarea dublurilor

void compactare(float x[], int\* n) {

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

int j = i + 1;

while (j < \*n) {

if (x[i] == x[j]) {

for (int k = j; k < \*n - 1; k++) {

x[k] = x[k + 1];

}

(\*n)--;

}

else {

j++;

}

}

}

}

//Subprogram care determina daca un numar este palindrom

int isPalindrom(int numar) {

int invers = 0, original, rest;

original = numar;

while (numar != 0) {

rest = numar % 10;

invers = invers \* 10 + rest;

numar = numar / 10;

}

if (original == invers) {

return 1;

}

else {

return 0;

}

}

// Subprogram care determina numarul de palindroame dintr-un vector

int nrPalindrom(int v[], int n) {

int numarPalindrom = 0;

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

if (isPalindrom(v[i])) {

numarPalindrom++;

}

}

return numarPalindrom;

}

int isPrim(int n) {

if (n < 2) {

return 0;

}

for (int i = 2; i \* i <= n; i++) {

if (n % i == 0) {

return 0;

}

}

return 1;

}

// Subprogram care determina numerele prime dintr-un vector

void numerePrime(int v[], int n) {

printf("Numerele prime din vector sunt: ");

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

if (isPrim(v[i])) {

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

}

}

printf("\n");

}

// Subprogram care realizeaza sortarea unui vector folosind algoritmul Buble Sort

void bubleSort(int v[], int n) {

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

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

if (v[j] > v[j + 1]) {

int aux = v[j];

v[j] = v[j + 1];

v[j + 1] = aux;

}

}

}

}

// Subprogram care primeste ca parametru un vector de numere intregi si returneaza un vector

// de aceeasi dimensiune care contine ultima cifra a fiecarui nr. din primul vector, ordinea elementelor ramane neschimbata

void ultimaCifra(int v[], int n, int rezultat[]) {

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

rezultat[i] = v[i] % 10;

}

}

// Subprogram pentru inversarea coloanelor si liniilor unei matrice intre ele

void inversareMatrice(int matrice[][100], int n, int m) {

int temp;

for (int i = 0; i < n / 2; i++) {

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

temp = matrice[i][j];

matrice[i][j] = matrice[n - i - 1][j];

matrice[n - i - 1][j] = temp;

}

}

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

for (int j = 0; j < m / 2; j++) {

temp = matrice[i][j];

matrice[i][j] = matrice[i][m - j - 1];

matrice[i][m - j - 1] = temp;

}

}

}

// Subprogram pentru ordonarea descrescatoare a liniilor unei matrice

void ordonareDescrescatoareLinii(int mat[][100], int n, int m) {

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

for (int j = 0; j < m - 1; j++) {

for (int k = 0; k < m - j - 1; k++) {

if (mat[i][k] < mat[i][k + 1]) {

int temp = mat[i][k];

mat[i][k] = mat[i][k + 1];

mat[i][k + 1] = temp;

}

}

}

}

}

// Subprogram care verifica daca o matrice este patrat magic

int estePatratMagic(int matrice[][100], int n) {

int sumaDiagonalaPrincipala = 0;

int sumaDiagonalaSecundara = 0;

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

sumaDiagonalaPrincipala += matrice[i][i];

sumaDiagonalaSecundara += matrice[i][n - i - 1];

}

if (sumaDiagonalaPrincipala != sumaDiagonalaSecundara) {

return 0;

}

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

int sumaLinie = 0;

int sumaColoana = 0;

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

sumaLinie += matrice[i][j];

sumaColoana += matrice[j][i];

}

if (sumaLinie != sumaDiagonalaPrincipala || sumaColoana != sumaDiagonalaPrincipala) {

return 0;

}

}

return 1;

}

int main() {

int optiune;

int n, m, value, aparitiiTarget, numarPalindrom;

float v[100], medie;

int w[100];

int matrice[100][100];

int rezultat[100];

do {

printf("Selectati optiunea:\n");

printf("1. Afisare vanzari zilnice\n");

printf("2. Afisare numar aparitii in vector\n");

printf("3. Compactare vector\n");

printf("4. Determinare numar de palindroame dintr-un vector\n");

printf("5. Determinare numere prime dintr-un vector\n");

printf("6. Sortare vector folosind Buble Sort\n");

printf("7. Ultima cifra a fiecarui numar dintr-un vector\n");

printf("8. Inversare coloane si linii matrice\n");

printf("9. Ordonare descrescatoare liniilor matrice\n");

printf("10. Verificare matrice patrat magic\n");

printf("0. Inchidere program\n");

scanf("%d", &optiune);

int n;

// Subprogram medie vanzari zilnice

float v[100], medie;

// Subprogram numar aparitii in vector

int target;

int w[100];

//Subprogram palindrom

int value;

//Subprograme matrice

int m;

int matrice[100][100];

switch (optiune) {

case 1:

n = citire(v);

medie = medieVanzariZilnice(v, n);

printf("Vanzarile zilnice sunt: ");

afisare(v, n);

if (medie) {

printf("\n Media vanzarilor este: %5.2f", medie);

}

else {

printf("\n Firma nu a inregistrat vanzari");

}

break;

case 2:

n = citire2(w);

afisare2(w, n);

printf("\nIntroduceti valoarea target pentru numarul de aparitii: ");

scanf("%d", &value);

aparitiiTarget = numarAparitii(w, n, value);

printf("\nNumarul de aparitii ale valorii %d in vector este: %d\n", value, aparitiiTarget);

break;

case 3:

n = citire(v);

compactare(v, &n);

afisare(v, n);

break;

case 4:

n = citire2(w);

numarPalindrom = nrPalindrom(w, n);

printf("Numarul de elemente palindrom din vector este: %d\n", numarPalindrom);

break;

case 5:

n = citire2(w);

numerePrime(w, n);

break;

case 6:

n = citire2(w);

printf("Vectorul inainte de sortare: ");

afisare2(w, n);

bubleSort(w, n);

printf("Vectorul dupa sortare: ");

afisare2(w, n);

break;

case 7:

n = citire2(w);

int rezultat[100];

ultimaCifra(w, n, rezultat);

printf("Ultima cifra a fiecarui numar: ");

afisare2(rezultat, n);

break;

case 8:

printf("Introduceti dimensiunile matricei (n m): ");

scanf("%d %d", &n, &m);

printf("Introduceti elementele matricei:\n");

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

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

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

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

}

}

printf("Matricea inainte de inversare:\n");

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

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

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

}

printf("\n");

}

inversareMatrice(matrice, n, m);

printf("Matricea dupa inversare:\n");

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

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

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

}

printf("\n");

}

break;

case 9:

printf("Introduceti dimensiunile matricei (n m): ");

scanf("%d %d", &n, &m);

printf("Introduceti elementele matricei:\n");

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

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

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

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

}

}

printf("Matricea inainte de ordonare:\n");

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

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

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

}

printf("\n");

}

ordonareDescrescatoareLinii(matrice, n, m);

printf("Matricea dupa ordonare:\n");

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

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

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

}

printf("\n");

}

break;

case 10:

printf("Introduceti dimensiunea matricei patratice (n): ");

scanf("%d", &n);

printf("Introduceti elementele matricei:\n");

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

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

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

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

}

}

if (estePatratMagic(matrice, n)) {

printf("Matricea este patrat magic.\n");

}

else {

printf("Matricea nu este patrat magic.\n");

}

break;

default:

printf("Optiune invalida");

}

} while (optiune != 0);

return 0;

}

