Ministerul Educaţiei, Culturii și Cercetării

al Republicii Moldova   
  
Universitatea Tehnică a Moldovei   
  
  
  
  
  
  
  
RAPORT

Lucrarea de laborator nr. 4   
la Programarea Calculatoarelor

A efectuat: st. gr. TI-211 Popa Cătălin   
  
A verificat: Stratulat Ștefan

UTM, Chișinău 2021

**Scopul lucrării**

Scopul lucrării este de a familiariza studentul cu scrierea și utilizarea tablourilor unidimensionale și bidimensionale, utilizînd limbajul C.

**Probleme pentru tablouri unidimensionale:**

**1. Scrieți un program C pentru a citi și a imprima elementele unui tablou. - folosind recursia.**

#include <stdio.h>

#include <stdlib.h>

void array(int arr[], int start, int n);

int main()

{

int n, i;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

printf("Elementele tabloului sunt : ");

array(arr, 0, n);

printf("\n\n");

return 0;

}

void array(int arr[], int start, int n)

{

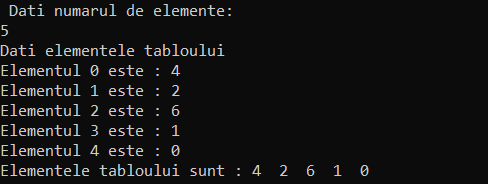
if(start >= n)

return;

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

array(arr, start+1, n);

}



**2. Scrieți un program C pentru a imprima toate elementele negative dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

void array(int arr[], int start, int n);

int main()

{

int n, i;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

array(arr, 0, n);

return 0;

}

void array(int arr[], int start, int n)

{

printf("Numerele negative sunt:\n");

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

{

if(arr[i]<0)

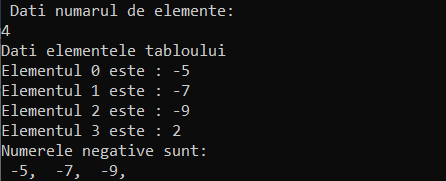
{

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

}

}

}



**3. Scrieți un program C pentru a găsi suma tuturor elementelor tabloului. - folosind recursia.**

#include <stdio.h>

#include <stdlib.h>

int sum(int arr[], int start, int n);

int main()

{

int n, i,s\_array;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

s\_array=sum(arr, 0, n);

printf("Suma elementelor este: %d\n",s\_array);

return 0;

}

int sum(int arr[], int start, int n)

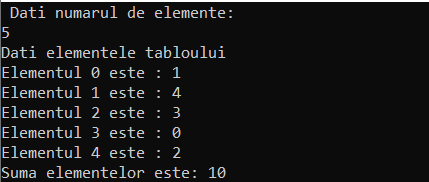
{

if(start >= n)

return 0;

return(arr[start] + sum(arr, start + 1, n));

}



**4. Scrieți un program C pentru a găsi elementul maxim și minim într-un tablou. - folosind recursia.**

#include <stdio.h>

#include <stdlib.h>

int maxnumber(int arr[], int start, int n);

int minnumber(int arr[], int start, int n);

int main()

{

int n, i,max,min;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

max = maxnumber(arr, 0, n);

min = minnumber(arr, 0, n);

printf("\n\n");

printf("Elementul maxim: %d\n",max);

printf("Elementul minim: %d\n",min);

return 0;

}

int maxnumber(int arr[], int start, int n)

{

int max;

if(start >= n-2)

{

if(arr[start] > arr[start + 1])

return arr[start];

else

return arr[start + 1];

}

max = maxnumber(arr, start + 1, n);

if(arr[start] > max)

return arr[start];

else

return max;

}

int minnumber(int arr[], int start, int n)

{

int min;

if(start >= n-2)

{

if(arr[start] < arr[start+1])

return arr[start];

else

return arr[start + 1];

}

min = minnumber(arr, start + 1, n);

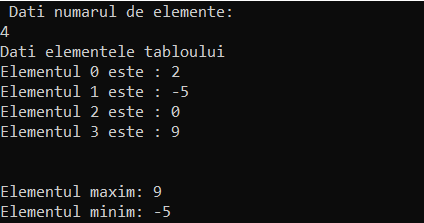
if(arr[start] < min)

return arr[start];

else

return min;

}



**5. Scrieți un program C pentru a găsi al doilea cel mai mare element dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,max,max2,i,j;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

//primul element maxim

max = 0;

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

{

if(max < arr[i])

{

max = arr[i];

j = i;

}

}

//al doilea element maxim

max2 = 0;

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

{

if(i == j)

{

i++;

i--;

}

else

{

if(max2 < arr[i])

{

max2 = arr[i];

}

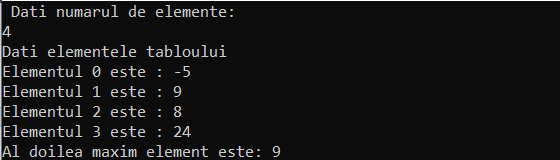
}

}

printf("Al doilea maxim element este: %d\n", max2);

return 0;

}



**6. Scrieți un program C pentru a număra numărul total de elemente pare și impare dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,s=0,a=0;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

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

{

if(arr[i]%2==0)

{

s+=1;

}

}

printf("Sunt: %d numere pare\n",s);

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

{

if(arr[i]%2 != 0)

{

a+=1;

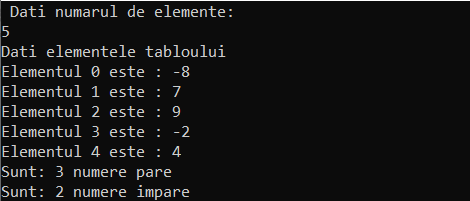
}

}

printf("Sunt: %d numere impare\n",a);

return 0;

}



**7. Scrieți un program C pentru a număra numărul total de elemente negative dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,s=0;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

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

{

if(arr[i] < 0)

{

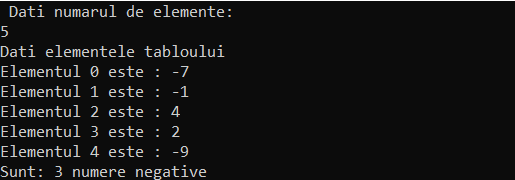
s+=1;

}

}

printf("Sunt: %d numere negative\n",s);

return 0;

}

**8. Scrieți un program C pentru a copia toate elementele dintr-un tablou în alt tablou.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

int arr2[100];

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

{

printf("Elementul %d este : ",i);

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

}

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

{

arr2[i] = arr[i];

}

printf("\n");

printf("Elementele primului tablou: \n");

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

{

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

}

printf("\n");

printf("Elementele altui tablou: \n");

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

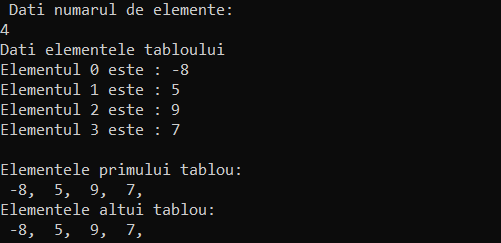
{

printf(" %d, ", arr2[i]);

}

return 0;

}



**9. Scrieți un program C pentru a insera un element într-un tablou pe o poziție dată de la tastatură.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,number,position;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

printf("Dati elementul care doriti sa il inserati: \n");

scanf("%d",&number);

printf("Dati pozitia pe care doriti sa il inserati: \n");

scanf("%d",&position);

for(i=n; i>=position; i--)

{

arr[i]=arr[i-1];

}

arr[position-1] = number;

n++;

printf("Tabelul dupa inserarea numarului: \n");

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

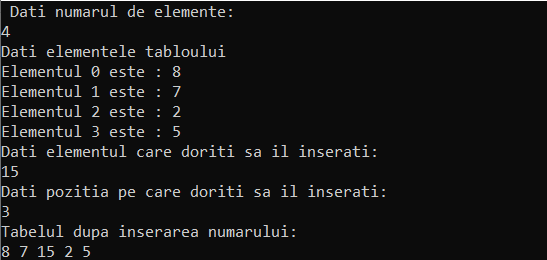
{

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

}

return 0;

}



**10. Scrieți un program C pentru a șterge un element dintr-un tablou în poziția specificată de la tastatură**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,position;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

printf("Dati pozitia elementului pe care doriti sa il stergeti: \n");

scanf("%d",&position);

for(i=position-1; i<n; i++)

arr[i] = arr[i+1];

printf("Rezultatul dupa stergere: \n");

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

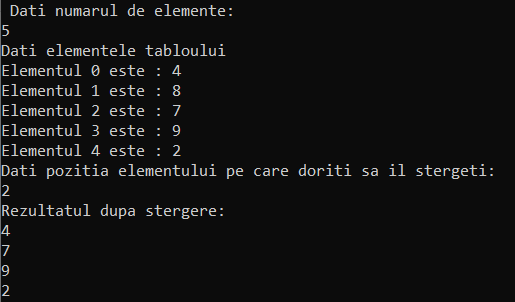
{

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

}

return 0;

}



**11. Scrieți un program C pentru a număra frecvența relativă fiecărui element dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

int main()

{

int n,i,j,k,f,s;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

int frec[100];

float frec2[100];

frec[100]=frec2[100];

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

{

printf("Elementul %d este : ",i);

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

frec[i] = -1;

}

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

{

f = 1;

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

{

if(arr[i] == arr[j])

{

f++;

frec[j] = 0;

}

}

if(frec[i]!= 0)

{

frec[i] = f;

}

}

printf("\n");

printf("Frecventa elementelor este: \n");

s=0;

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

{

if(frec[i]!= 0)

{

printf("Frecventa elementului |%d| este : %d \n", arr[i], frec[i]);

s+=frec[i];

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

{

frec2[i] = frec[i];

}

}

}

printf("\n\n");

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

{

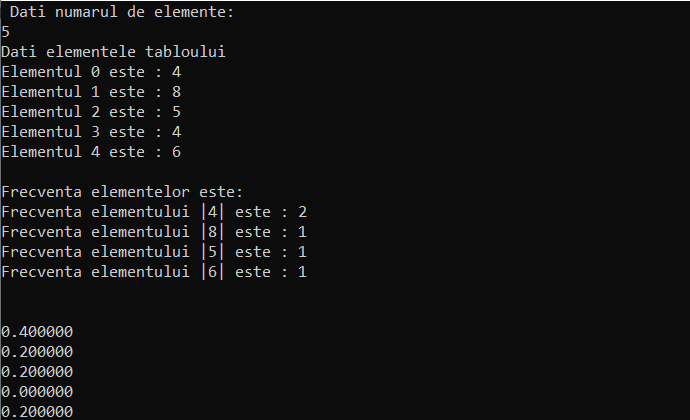
frec2[i]=frec2[i]/(float)s;

printf("%f \n",frec2[i]);

}

return 0;

}



**12. Scrieți un program C pentru a imprima toate elementele unice din tablou.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

int main()

{

int n,i,j,k,a;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

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

{

printf("Elementul %d este : ",i);

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

}

printf("\n\n");

printf("Elementele unice sunt:\n");

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

{

a=0;

for(j=0,k=n; j<k+1; j++)

{

if (i!=j)

{

if(arr[i]==arr[j])

{

a++;

}

}

}

if(a==0)

{

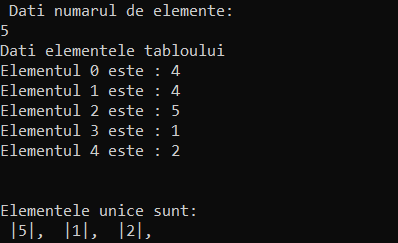
printf(" |%d|, ",arr[i]);

}

}

return 0;

}



**13. Scrieți un program C pentru a număra numărul total de elemente duplicate dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

int main()

{

int n,i,j,a=1,b=0;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100];

int arr2[100];

int arr3[100];

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

{

printf("Elementul %d este : ",i);

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

}

printf("\n\n");

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

{

arr2[i]=arr[i];

arr3[i]=0;

}

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

{

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

{

if(arr[i]==arr2[j])

{

arr3[j]=a;

a++;

}

}

a=1;

}

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

{

if(arr3[i]==2)

{

b++;

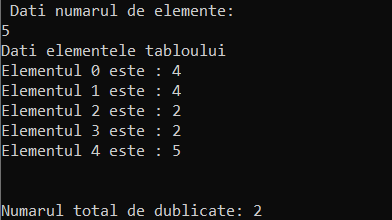
}

}

printf("Numarul total de dublicate: %d \n", b);

return 0;

}



**14. Scrieți un program C pentru a șterge toate elementele duplicate dintr-un tablou.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

int main()

{

int n,i,j,a=0;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

int arr[100],arr2[100];

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

{

printf("Elementul %d este : ",i);

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

}

printf("\n\n");

printf("Elementele inainte de stergere : ");

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

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

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

{

int j;

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

{

if (arr[i] == arr2[j])

break;

}

if (j == a)

{

arr2[a] = arr[i];

a++;

}

}

printf("\n");

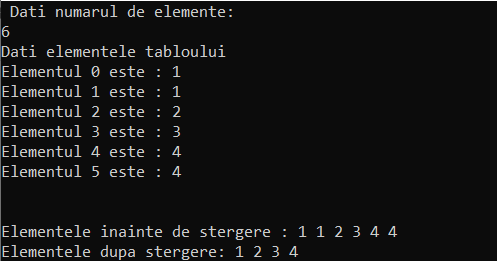
printf("Elementele dupa stergere: ");

for (int i = 0; i < a; i++)

printf("%d ", arr2[i]);

return 0;

}



**15. Scrieți un program C pentru a îmbina două tablouri în al treilea tablou.**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

tabel(int \*a, int n)

{

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

{

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

}

}

int main()

{

int n,m,i,arr[100],arr2[100],arr3[100];

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

printf(" Dati numarul de elemente ale tabloului 2: \n");

scanf("%d",&m);

printf("Dati elementele tabloului 2\n");

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

{

printf("Elementul %d este : ",i);

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

}

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

{

if(i < n)

arr3[i] = arr[i];

else

arr3[i] = arr2[i-m];

}

printf("\nElementele primului tablou: \n");

tabel(arr,n);

printf("\nElementele celui de-a doilea tablou: \n");

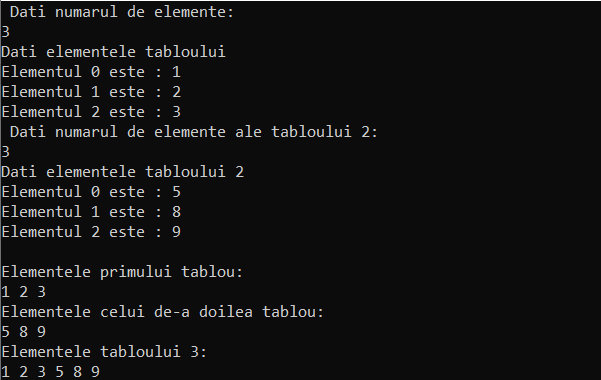
tabel(arr2,m);

printf("\nElementele tabloului 3: \n");

tabel(arr3,n+m);

return 0;

}



**16. Scrieți un program C pentru a inversa ordinea elementelor unui tablou.**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

int main()

{

int n,i,arr[100];

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

printf("\nTabelul inversat este: \n");

for(i=n-1; i>=0; i--)

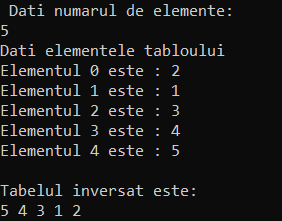
{

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

}

return 0;

}



**17. Scrieți un program C pentru a pune elemente pare și impare ale unui tablou în două tablouri separate.**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

tabel(int \*a, int n)

{

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

{

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

}

}

int main()

{

int n,m,l,i,j,k,arr[100],arr2[100],arr3[100];

j=k=0;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

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

{

if(arr[i]%2 == 0)

arr2[j++] = arr[i];

else

arr3[k++] = arr[i];

}

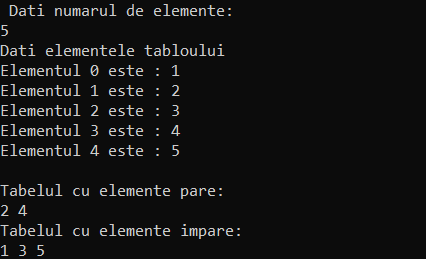
printf("\nTabelul cu elemente pare: \n");

tabel(arr2,j);

printf("\nTabelul cu elemente impare: \n");

tabel(arr3,k);

return 0; }



**18. Scrieți un program C pentru a căuta un element din tablou și afiza pozițiile găsite.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,e,a=0,arr[10];

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

printf("Dati elementul pe care doriti sa il gasiti: \n");

scanf("%d",&e);

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

{

if(arr[i] == e)

{

a=1;

break;

}

}

if(a==1)

{

printf("Elementul %d se afla pe pozitia: %d", e, i+1);

}

else

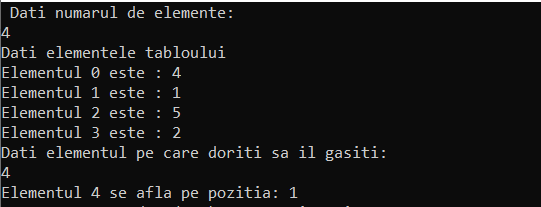
{

printf("Elementul dat nu se afla in tablou\n");

}

return 0;

}



**19. Scrieți un program C pentru a sorta elementele tabloului în ordine crescătoare sau descrescătoare.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,j,a,a2,arr[100],b,d;

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

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

{

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

{

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

{

a=arr[i];

arr[i]=arr[j];

arr[j]=a;

}

}

}

printf("\nCrescator: \n");

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

{

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

}

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

{

for(d=b+1; d<n; d++)

{

if(arr[b]<arr[d])

{

a2=arr[b];

arr[b]=arr[d];

arr[d]=a2;

}

}

}

printf("\nDescrescator: \n");

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

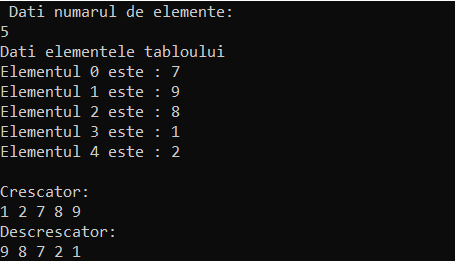
{

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

}

return 0;

}



**20. Scrieți un program C pentru a sorta separat elementele pare și impare ale tabloului.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,j,k=0,a,b=0,arr[100],arr2[100];

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

if(arr[i]%2 == 1)

{

b++;

}

}

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

{

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

{

if(arr[j]>arr[j+1])

{

a=arr[j];

arr[j]=arr[j+1];

arr[j+1]=a;

}

}

}

j = n-b;

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

{

if(arr[i]%2 == 0)

{

if(k<n-b)

{

arr2[k++]=arr[i];

}

}

else

{

if(j<n)

{

arr2[j++]=arr[i];

}

}

}

printf("\nElementele dupa sortare: \n");

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

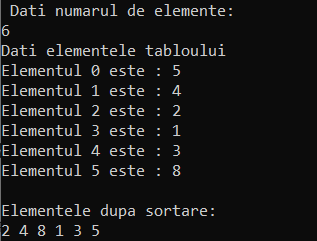
{

arr[i]=arr2[i];

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

}

return 0; }



**21. Scrieți un program C pentru a roti la stânga un tablou, operația ROL.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,j,a,k=1,arr[100];

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

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

{

a=arr[0];

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

{

arr[j] = arr[j+1];

}

arr[j] = a;

}

printf("\nTabelul dupa rotire la stanga: \n");

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

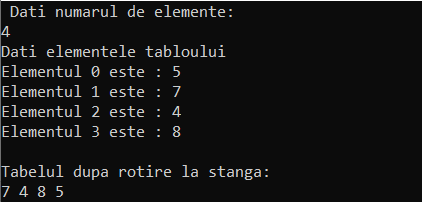
{

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

}

return 0;

}



**22. Scrieți un program C pentru a roti dreapta un tablou, operația ROR.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,i,j,a,k=1,arr[100];

printf(" Dati numarul de elemente: \n");

scanf("%d",&n);

printf("Dati elementele tabloului\n");

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

{

printf("Elementul %d este : ",i);

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

}

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

{

a=arr[n-1];

for(j=n-1; j>0; j--)

{

arr[j] = arr[j-1];

}

arr[0] = a;

}

printf("\nTabelul dupa rotire: \n");

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

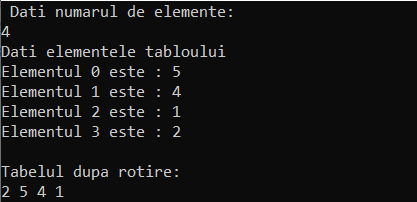
{

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

}

return 0;

}



**2. Probleme pentru tablouri bidimensionale:**

**1. Scrieți un program C pentru a aduna două matrice.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100], sum[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

printf("Dati elementul arr[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

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

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

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

{

printf("Dati elementul arr2[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

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

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

{

sum[i][j] = arr[i][j] + arr2[i][j];

}

printf("\n Suma matricelor este: \n");

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

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

{

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

if (j == m - 1)

{

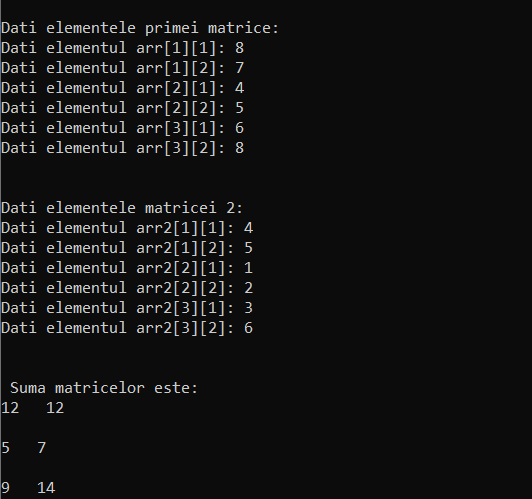
printf("\n\n");

}

}

return 0;

}



**2. Scrieți un program C pentru a scădea două matrice.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100], dif[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

printf("Dati elementul arr[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

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

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

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

{

printf("Dati elementul arr2[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

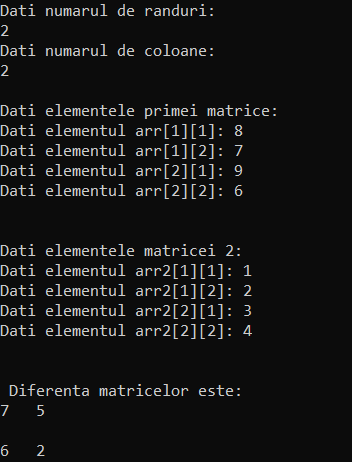
for (i = 0; i < n; ++i)

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

{

dif[i][j] = arr[i][j] - arr2[i][j];

}

 printf("\n Diferenta matricelor este: \n");

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

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

{

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

if (j == m - 1)

{

printf("\n\n");

}

}

return 0;

}

**3. Scrieți un program C pentru a efectua multiplicarea matricei cu un scalar.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,m,i,j,num,arr[100][100], arr2[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

printf("Dati elementul arr[%d][%d]: ", i + 1, j + 1);

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

}

printf("Dati multiplicatorul: \n");

scanf("%d",&num);

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

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

{

arr2[i][j] = arr[i][j] \* num;

}

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

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

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

{

printf("%d ", arr2[i][j]);

if(j == m-1)

{

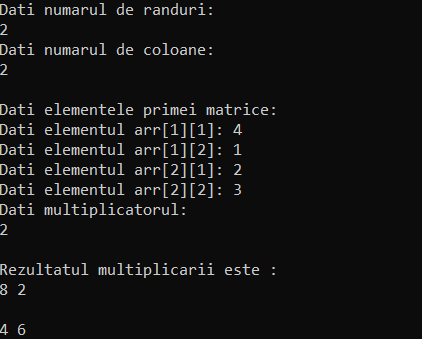
printf("\n\n");

}

}

return 0;

}



4. Scrieți un program C pentru a multiplica două matrice.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100],multi[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

printf("Dati elementul arr[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

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

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

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

{

printf("Dati elementul arr2[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

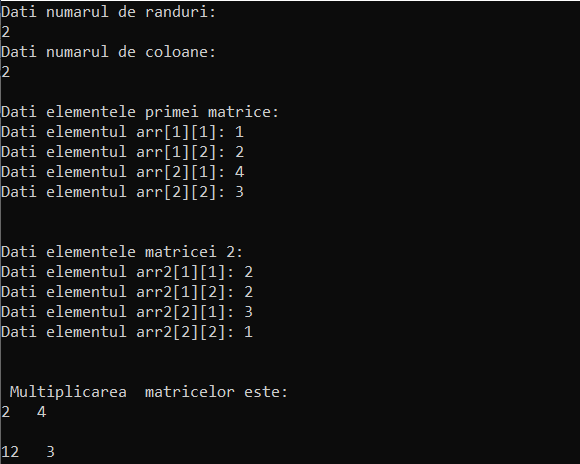
for(i=0; i<n; i++)

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

{

multi[i][j] = arr[i][j] \* arr2[i][j];

}

 printf("\n Multiplicarea matricelor este: \n");

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

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

{

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

if (j == m - 1)

{

printf("\n\n");

}

}

return 0;

}

**5. Scrieți un program C pentru a verifica dacă două matrice sunt egale sau nu.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100],num;

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

printf("Dati elementul arr[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

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

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

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

{

printf("Dati elementul arr2[%d][%d]: ", i + 1, j + 1);

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

}

printf("\n");

num=1;

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

{

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

{

if(arr[i][j]!=arr2[i][j])

{

num=0;

break;

}

}

}

if(num==1)

{

printf("Matricele sunt egale\n");

}

else

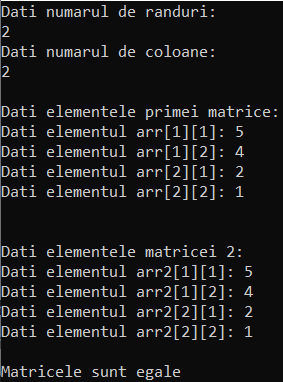
{

printf("Matricele nu sunt egale\n");

}

return 0;

}



**6. Scrieți un program C pentru a găsi suma elementelor diagonale principale ale unei matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100],s=0;

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

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

{

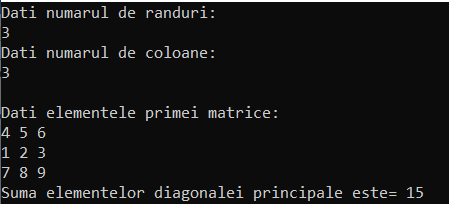
s+=arr[i][i];

}

printf("Suma elementelor diagonalei principale este= %d",s);

return 0;

}



**8. Scrieți un program C pentru a găsi suma fiecărui rând și coloană a unei matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void randsuma(int arr[100][100], int m, int n);

void colsuma(int arr[100][100], int m, int n);

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100],s=0;

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

printf("\nSuma randurilor: \n");

randsuma(arr,m,n);

printf("\nSuma coloanelor: \n");

colsuma(arr,m,n);

return 0;

}

void randsuma(int arr[100][100], int m, int n)

{

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

{

int rs=0;

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

{

rs+=arr[i][j];

}

printf("\nSuma elementelor pe randul %d este : %d\n",i+1,rs);

}

}

void colsuma(int arr[100][100], int m, int n)

{

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

{

int cs=0;

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

{

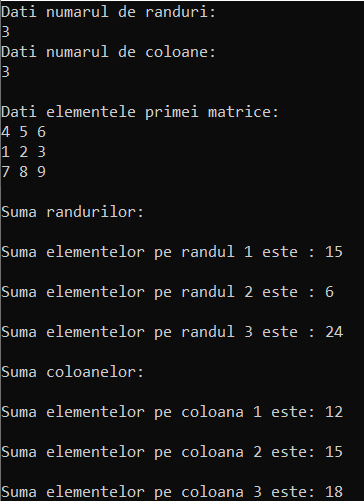
cs+=arr[j][i];

}

printf("\nSuma elementelor pe coloana %d este: %d\n",i+1,cs);

}

}



**9. Scrieți un program C pentru a schimba diagonalele unei matrice.**

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100], arr2[100][100],a;

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

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

{

a=arr[i][i];

arr[i][i]=arr[i][n-i-1];

arr[i][n-i-1]=a;

}

printf("\n");

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

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

{

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

if (j == m - 1)

{

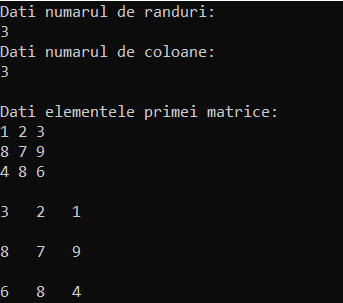
printf("\n\n");

}

}

return 0;

}



**10. Scrieți un program C pentru a găsi matricea triunghiulară superioară.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

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

{

printf("\n");

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

{

if(j>=i)

{

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

}

else

{

printf("0 ");

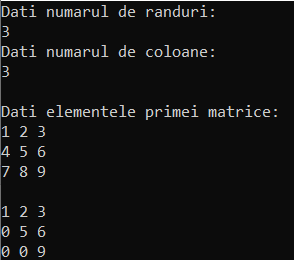
}

}

}

return 0;

}



**11. Scrieți un program C pentru a găsi matricea triunghiulară inferioară.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

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

{

printf("\n");

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

{

if(i>=j)

{

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

}

else

{

printf("0 ");

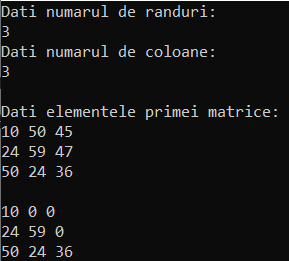
}

}

}

return 0;

}



**12. Scrieți un program C pentru a găsi suma matricei triunghiulare superioare.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

if(n==m)

{

int sum=0;

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

{

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

{

if(i<=j)

sum+=arr[i][j];

}

}

printf("Suma matricei triunghiulare superioare: %d\n",sum);

}

else

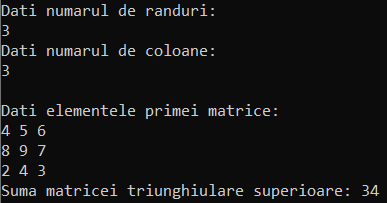
{

printf("Matricea nu are structura necessara\n");

}

return 0;

}



**13. Scrieți un program C pentru a găsi suma matricei triunghiulare inferioare.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

if(n==m)

{

int sum=0;

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

{

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

{

if(j<=i)

sum+=arr[i][j];

}

}

printf("Suma matricei triunghiulare inferioare: %d\n",sum);

}

else

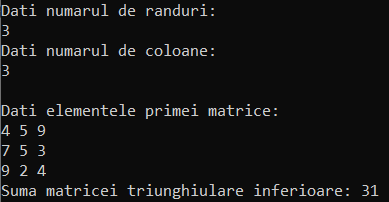
{

printf("Matricea nu are structura necessara\n");

}

return 0;

}



**14. Scrieți un program C pentru a găsi transpunerea unei matrice.**

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[100][100],arr2[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

printf("Matricea initiala: \n");

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

{

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

{

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

if(j==m-1)

printf("\n");

}

}

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

{

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

{

arr2[j][i]=arr[i][j];

}

}

printf("Matricea transpusa:\n");

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

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

{

printf("%d ", arr2[i][j]);

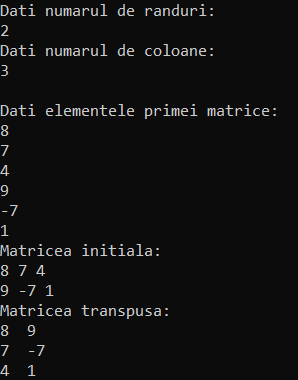
if (j == n - 1)

printf("\n");

}

return 0;

}



**15. Scrieți un program C pentru a găsi determinantul unei matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,i,j,arr[3][3],a,b,c,det;

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

printf("\nDati elementele primei matrice:\n");

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

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

{

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

}

a = (arr[1][1] \* arr[2][2]) - (arr[2][1] \* arr[1][2]);

b = (arr[1][0] \* arr[2][2]) - (arr[2][0] \* arr[1][2]);

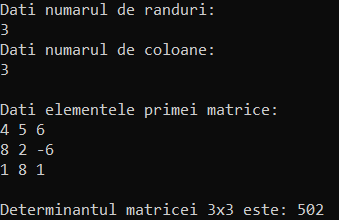
c = (arr[1][0] \* arr[2][1]) - (arr[2][0] \* arr[1][1]);

det = (arr[0][0] \* a) - (arr[0][1] \* b) + (arr[0][2] \* c);

printf("\nDeterminantul matricei 3x3 este: %d\n",det);

return 0;

}



**16. Scrieți un program C pentru a verifica dacă matricea este unitară.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

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

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

printf("\n");

printare(arr,n,m);

printf("\n");

unitara(arr,n,m);

}

void unitara(int arr[100][100],int n, int m)

{

int i,j,a=0;

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

{

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

{

if(i==j && arr[i][j]!=1)

{

a=-1;

break;

}

else if(i!=j && arr[i][j]!=0)

{

a=-1;

break;

}

}

}

if(a==0)

{

printf("Matricea este unitara\n");

}

else

{

printf("Matricea nu este unitara\n");

}

}

void printare(int arr[100][100],int n, int m)

{

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

{

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

{

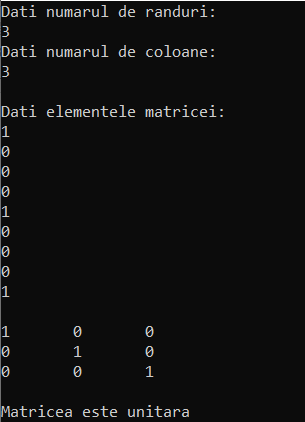
printf("%d\t",arr[i][j]);

}

printf("\n");

}

}



**17. Scrieți un program C pentru a verifica dacă matrix este rarefiată (dacă numărul de zerouri reprezintă mai mult sau egal cu 70% din elementele matricei, atunci matricea este rarefiată).**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

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

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

printf("\n");

printare(arr,n,m);

printf("\n");

rarefiata(arr,n,m);

}

void rarefiata(int arr[100][100],int n, int m)

{

int i,j;

double a=0,n1,m1,ver;

n1=n;

m1=m;

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

{

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

{

if(arr[i][j] == 0)

{

a++;

}

}

}

printf("\n");

ver=(a/(n1\*m1))\*100;

if(ver >= 70)

{

printf("Matricea este rarefiata\n");

}

else

{

printf("Matricea nu este rarefiata\n");

}

}

void printare(int arr[100][100],int n, int m)

{

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

{

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

{

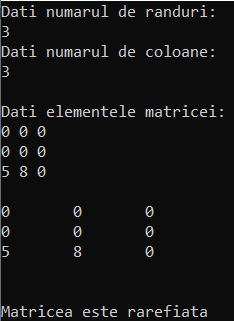
printf("%d\t",arr[i][j]);

}

printf("\n");

}

}



**18. Scrieți un program C pentru a verifica dacă matricea este simetrică.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,arr[100][100],arr2[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

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

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

{

arr2[i][j]=arr[i][j];

}

printf("\n");

printare(arr,n,m);

printf("\n");

printare(arr2,n,m);

printf("\n");

simetrica(arr,arr2,n,m);

}

void simetrica(int arr[100][100],int arr2[100][100],int n, int m)

{

int i,j,s=0;

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

{

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

{

arr[i][j] == arr[j][i];

}

}

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

{

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

{

if(arr[i][j] == arr2[j][i])

{

s=0;

}

else

{

s++;

}

}

}

if(s == 0)

{

printf("Matricea este simetrica\n");

}

else

{

printf("Matricea nu este simetrica\n");

}

}

void printare(int arr[100][100],int n, int m)

{

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

{

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

{

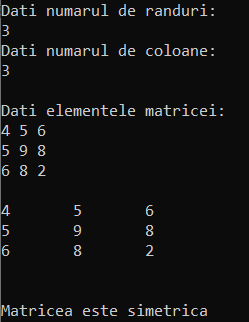
printf("%d\t",arr[i][j]);

}

printf("\n");

}

}



**19. Scrieți un program C pentru a roti la stînga o matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

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

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

printf("\n");

printare(arr,n,m);

printf("\n");

rotares(arr,n,m);

}

void rotares(int arr[100][100],int n, int m)

{

int i,j;

for(int i=n-1; i>=0; i--)

{

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

{

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

}

printf("\n");

}

}

void printare(int arr[100][100],int n, int m)

{

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

{

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

{

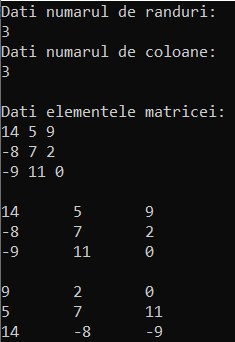
printf("%d\t",arr[i][j]);

}

printf("\n");

}

}



**20. Scrieți un program C pentru a roti la dreapta o matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

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

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

printf("\n");

printare(arr,n,m);

printf("\n");

rotared(arr,n,m);

}

void rotared(int arr[100][100],int n, int m)

{

int i,j;

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

{

for(int j=m-1; j>=0; j--)

{

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

}

printf("\n");

}

}

void printare(int arr[100][100],int n, int m)

{

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

{

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

{

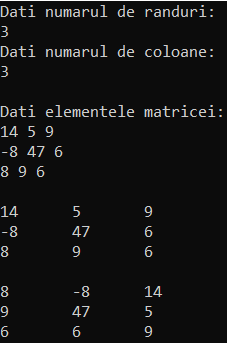
printf("%d\t",arr[i][j]);

}

printf("\n");

}

}



**21. Scrieți un program C pentru a roti în sus o matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,m,arr[100][100],arr2[100][100];

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

printf("\n");

printare(arr,n,m);

printf("\n");

rotared(arr,arr2,n,m);

printare(arr2,n,m);

}

void rotared(int arr[100][100],int arr2[100][100],int n, int m)

{

int i,j;

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

{

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

{

if(i == 0)

{

arr2[n-1][j] = arr[i][j];

}

else

{

arr2[i-1][j] = arr[i][j];

}

}

}

}

void printare(int arr[100][100],int n, int m)

{

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

{

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

{

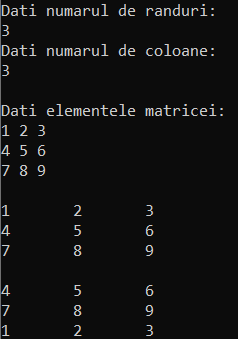
printf("%d\t",arr[i][j]);

}

printf("\n");

}

}



**22. Scrieți un program C pentru a roti în jos o matrice.**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

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

printf("Dati numarul de randuri: \n");

scanf("%d", &n);

printf("Dati numarul de coloane: \n");

scanf("%d", &m);

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

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

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

{

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

}

printf("\n");

printare(arr,n,m);

printf("\n");

rotared(arr,n,m);

printare(arr,n,m);

}

void rotared(int arr[100][100],int n, int m)

{

int i,j,a;

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

{

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

{

a=arr[i][j];

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

arr[n-1][j]=a;

}

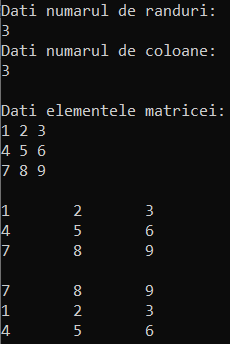
}

}

void printare(int arr[100][100],int n, int m)

{

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

 {

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

{

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

}

printf("\n");

}

}

**Intrebari de control:**

* ce reprezintă un tablou unidimensional în limbajul C?

Un sir de elemente de acelasi tip de date,aranjate intr-un rand.

* ce reprezintă un tablou bidimensional în limbajul C?

Reprezinta o matrice,avand elemente de acelasi tip de date, dispuse de linii si coloane.

* cu ce operator se accesează elementele unui tablou în limbajul C?

Elementele unui tablou se acceseazaz prin operatorul de indexare [].

* ce tip de date pot fi indecșii unui tablou în limbajul C?

Indecsii unui tablou in limbajul C poate fi tipul de date int-integer.

* ce index are primul element din tabloul unidimensional ?

Primul element din tabloul unidimensional are indexul 0.

* ce index are ultimul element din tabloul unidimensional cu lungimea N?

Ultimul element din tabloul unidimensional cu lungimea N are indexul N-1.

* ce algoritmi de sortare a tablourilor ați utilizat în lucrare?

1.Metoda sortarii prin selectie directa.

2.Sortarea prin metoda bulelor.