

## Lucrarea de laborator N 11

### Tema: "Meniuri în C++"

Condițiile din fiecare variantă includ câte un masiv bidimensional  $x[n,n]$  care conține o arie hașurată în mod specific pentru fiecare variantă în parte. Rezolvarea problemei se reduce la prelucrarea elementelor din ariile hașurată și nehașurată, conform condițiilor variantei.

Primul pas în rezolvarea sarcinii va fi afișarea condițiilor problemei la monitor în mod program. Completarea masivului, adică atribuirea valorilor elementelor masivului va fi efectuată manual sau automat la alegerea utilizatorului.

După completarea masivului, acesta va fi afișat într-o formă comodă de citit, adică elementele fiind ordonate pe linii și coloane. Mai mult ca atât, masivul va fi afișat în două culori: elementele ariei hașurate într-o culoare, iar elementele ariei nehașurate în altă culoare.

Apoi urmează rezolvarea fiecărui din cele trei puncte ale sarcinii. Pentru acest scop vor fi create câteva subprograme, fiecare din ele rezolvând o subsarcină concretă. Cantitatea de subprograme va depinde de condițiile problemei și dorința executorului.



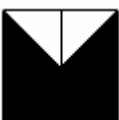


După rezolvarea celor trei puncte ale sarcinii și afișarea rezultatelor primite masivul final va fi afișat la monitor.







Cu scopul de a face mai comodă utilizarea programului, în cadrul lui va fi creat un meniu, cu ajutorul cărui va fi simplu de trecut dintr-o etapă a rezolvării în alta. În așa fel programul trebuie să conțină următoarele puncte de meniu:







- 1) afișarea condițiilor problemei;
- 2) completarea masivului în mod manual;
- 3) completarea masivului în mod automat;
- 4) afișarea masivului (în 2 culori);
- 5) rezolvarea punctului a) din condiție;
- 6) rezolvarea punctului b) din condiție;
- 7) rezolvarea punctului c) din condiție;
- 8) afișarea masivului final (în 2 culori);
- 9) ieșire.


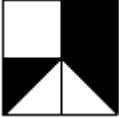

După rezolvarea fiecărui punct din cele trei existente, rezultatul corespunzător va fi afișat la monitor. Punctele din meniu 5,6,7 ce corespund rezolvării punctelor a,b,c din condiție pot fi unite într-un număr mai mic sau despărțite într-un număr mai mare de puncte de meniu, în dependență de condițiile problemei și dorința executorului.

Variante

Nr. var.	Condiția	Masivul
1.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat media aritmetică a elementelor impare din aria hașurată.</p> <p>b) De aflat poziția și cantitatea elementelor din masiv egale cu cifra propusă de utilizator.</p> <p>c) Dacă element egal cu cifra data nu există în masiv, de înscris într-un fișier un comentariu respectiv și de generat un semnal audio.</p>	
2.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat elementele maxime din fiecare porțiune a ariei hașurate cu vecinii lor.</p> <p>b) De calculat media aritmetică a elementelor pare din toată aria hașurată.</p> <p>c) De înlocuit toate elementele din aria nehașurată cu zero.</p>	
3.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat suma primului și ultimului element par din aria hașurată și poziția lor.</p> <p>b) De aflat câte elemente cu valoarea acestei sume sînt în aria nehașurată.</p> <p>c) De scris în fișier valorile și pozițiile tuturor vecinilor celui mai mic element pozitiv din aria hașurată.</p>	
4.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat media aritmetică a elementelor divizibile la patru din aria hasurată.</p> <p>b) De egalat toate elementele din aria nehasurată cu zero de culoare galbenă.</p> <p>c) De aflat valoarea și locul elementelor maxime din fiecare rînd din aria hașurată.</p>	
5.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De prezentat elementele pare cu vecinii lor din aria hașurată (în valori și poziții).</p> <p>b) Elementele negative din aria nehașurată de înlocuit cu "8".</p> <p>c) De scris în fișier numărul elementelor impare din aria hașurată.</p>	

6.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) Aflați media aritmetică a elementelor <math>&gt;5</math> din aria hașurată.</p> <p>b) Scrieți în fișier cantitatea elementelor din aria hașurată egale cu cifra utilizatorului.</p> <p>c) Aflați suma elementelor de pe perimetrul ariei nehașurate.</p>	
7.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) Aflați media aritmetică a elementelor pare din aria hașurată și o înscriteți în fișier.</p> <p>b) Aflați elementul maximal de pe fiecare coloană pară din aria hașurată.</p> <p>c) Aflați câte elemente în aria hașurată sînt egale cu cifra user-ului.</p>	
8.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) Aflați media aritmetică a elementelor de pe perimetrul ariei hașurate.</p> <p>b) Aflați numărul elementelor pare din fiecare coloană impară din aria hașurată.</p> <p>c) Aflați cel mai mare element din fiecare porțiune nehașurată și înscriteți-le în fișier.</p>	
9.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat valoarea și poziția fiecărui element par din aria hașurată.</p> <p>b) Aranjați în ordine descrescătoare elementele din fiecare coloană din aria hașurată.</p> <p>c) De aflat numărul elementelor pare din aria " * ".</p>	
10.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) Aranjați în ordine crescătoare elementele fiecărui rînd din aria hașurată.</p> <p>b) Aflați elementele maximale de pe fiecare coloană din aria hașurată.</p> <p>c) Aflați cantitatea elementelor din aria nehașurată egale cu cifra utilizatorului și scrieți-o într-un fișier.</p>	
11.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat poziția și cantitatea elementelor egale cu cifra propusă de utilizator.</p> <p>b) Aflați elementul maximal de pe fiecare coloană impara din aria hașurată.</p>	

	c) Înscrieți într-un fișier suma elementelor negative din masiv.	
12.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De calculat produsul elementelor impare din aria hașurată.</p> <p>b) Determinați elementul maximal de pe perimetrul ariei hașurate.</p> <p>c) Înscrieți într-un fișier valorile și pozițiile elementelor din fiecare colț al ariei hașurate.</p>	
13.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat media aritmetică a elementelor divizibile la trei din aria hașurată.</p> <p>b) Aflați cantitatea elementelor impare din fiecare coloană pară din aria hașurată.</p> <p>c) Înscrieți în fișier cantitatea zerourilor din aria nehașurată.</p>	
14.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat câte elemente în aria hașurată sînt egale cu cifra user-ului.</p> <p>b) De aflat valoarea și poziția elementelor minimale din fiecare rînd din aria hașurată.</p> <p>c) Înscrieți în fișier suma elementelor pare din aria nehașurată.</p>	
15.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) Aranjați în ordine crescătoare elementele din fiecare coloană din aria hașurată.</p> <p>b) Aflați cantitatea elementelor din aria nehașurată egale cu cifra utilizatorului.</p> <p>c) Înscrieți în fișier produsul elementelor impare din aria nehașurată.</p>	
16.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat media aritmetică a elementelor pare din aria hașurată.</p> <p>b) De prezentat elementele pare cu vecinii lor din aria hașurată (în valori și poziții).</p> <p>c) De comparat sumele elementelor din ariile hașurată și nehașurată.</p>	
17.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat suma primului și ultimului element par din aria hașurată și poziția lor.</p> <p>b) Aflați suma elementelor de pe perimetrul ariei nehașurate.</p> <p>c) Comparați produsele elementelor impare din ariile hașurată și nehașurată.</p>	

18.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De aflat suma primului și ultimului element impar din aria hașurată și poziția lor.</p> <p>b) Aflați suma elementelor de pe perimetrul ariei nehașurate.</p> <p>c) Comparați produsele elementelor impare din ariile hașurată și nehașurată</p>	
19.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) De scris în fișier valorile și pozițiile tuturor vecinilor celui mai mare element pozitiv din aria hașurată.</p> <p>b) Aflați media aritmetică a elementelor de pe perimetrul ariei hașurate.</p> <p>c) Comparați mediile aritmetice ale elementelor pozitive din ariile hașurată și nehașurată</p>	
20.	<p>Este dat masivul <math>x[n][n]</math>.</p> <p>a) Aranjați în ordine descrescătoare elementele fiecărui rând din aria hașurată.</p> <p>b) De egalat toate elementele din aria nehașurată cu zero de culoare galbenă.</p> <p>c) Comparați mediile aritmetice ale elementelor pozitive din ariile hașurată și nehașurată</p>	

### Exemplu de rezolvare a unei sarcini

Este dat masivul bidimensional  $x[n][n]$  cu aria hașurată

- a) De aflat valorile și pozițiile elementelor maxime din cele 2 porțiuni ale ariei hașurate.
- b) De înscris într-un fișier suma elementelor pare din aria hașurată.
- c) De aranjat elementele fiecărui rând din aria hașurată în ordine crescătoare.



Listinul programului:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<math.h>
```

```
#include<stdlib.h>
```

```
//afisarea conditiilor problemei la monitor
```

```
void conditii (void) {
```

```

int i,j,n=10,k; clrscr();
gotoxy(20,2);textcolor(11);
cprintf("\n");
gotoxy(29,4);textcolor(15); cprintf("Conditiiile problemei:");
gotoxy(10,5);textcolor(11);
cprintf("Este dat masivul bidimensional x[N][N] cu aria hasurata (rosie)");
gotoxy(5,6); cprintf("a) De aflat valorile si pozitiile elementelor maxime");
gotoxy(5,7); cprintf(" din cele 2 portiuni ale ariei hasurate"); gotoxy(5,8);
cprintf("b) De inscris intr-un fisier suma elementelor pare din aria hasurata");
gotoxy(5,9);
cprintf("c) De aranjat fiecare rind din aria hasurata in ordine crescatoare");
k=floor(float(n)/float(2));
for(i=0;i<n;i++) {
for(j=0;j<n;j++) {
if( (i<k) && ( j>=i) && (j<(n-i)) )
{gotoxy(15+(j*3),11+i);textcolor(12); cprintf("* ");}
else if( (i>=k) && ( j>=(n-i-1)) && (j<=i) )
{gotoxy(15+(j*3),11+i);textcolor(12);cprintf("* ");}
else {gotoxy(15+(j*3),11+i);textcolor(10);cprintf("* ");} } }
textcolor(15);gotoxy(10,wherey()+2);
cprintf("Pentru iesire in meniu culegeti <ENTER>"); getch(); }
//afisarea masivului initial in 2 culori la monitor
void afisare (int y[50][50],int m) {
int i,j,k; clrscr();gotoxy(15,2);textcolor(15);
cprintf("Masivul x[%d][%d] este:\n",m,m); k=floor(float(m)/float(2));
for(i=0;i<m;i++) {
for(j=0;j<m;j++) {
if( (i<k) && ( j>=i) && (j<(m-i)) )
{gotoxy(15+(j*4),4+i);textcolor(12); cprintf("%3d ",y[i][j]);}
else if( (i>=k) && ( j>=(m-i-1)) && (j<=i) )
{gotoxy(15+(j*4),4+i);textcolor(12);cprintf("%3d ",y[i][j]);}
else {gotoxy(15+(j*4),4+i);textcolor(10);cprintf("%3d ",y[i][j]);} } }
textcolor(15);gotoxy(10,wherey()+2);
cprintf("Pentru iesire in meniu culegeti <ENTER>"); getch(); }
//determinarea elementelor maxime in valori si pozitii din cele 2 portiuni hasurate
void maxim (int y[50][50],int m) {
int i,j,k,max1,rmax1,cmax1,max2,rmax2,cmax2;
//portiunea de sus a ariei hasurate.
k=floor(float(m)/float(2));

```

```

max1=y[0][0]; rmax1=0; cmax1=0;
for(i=0;i<k;i++) {
for(j=i;j<(m-i);j++) {
if (y[i][j]>=max1) {max1=y[i][j]; rmax1=i; cmax1=j;} } }
//portiunea de jos a ariei hasurate.
max2=y[m-1][0]; rmax2=m-1; cmax2=0;
for(i=k;i<m;i++) {
for(j=(m-i-1);j<=i;j++) {
if (y[i][j]>=max2) {max2=y[i][j]; rmax2=i; cmax2=j;} } }
//afisarea rezultatelor
clrscr();gotoxy(15,2);textcolor(15);
printf("Elementele maximele din ariile hasurate:");
for(i=0;i<m;i++) {
for(j=0;j<m;j++) {
if( (i<k) && ( j>=i) && (j<(m-i)) )
{gotoxy(15+(j*4),4+i);textcolor(12); cprintf("%3d ",y[i][j]);}
else if( (i>=k) && ( j>=(m-i-1)) && (j<=i) )
{gotoxy(15+(j*4),4+i);textcolor(12);cprintf("%3d ",y[i][j]);}
else {gotoxy(15+(j*4),4+i);textcolor(10);cprintf("%3d ",y[i][j]);}
if( ((i==rmax1)&&(j==cmax1)) || ((i==rmax2)&&(j==cmax2)) )
{gotoxy(15+(j*4),4+i);textcolor(15);cprintf("%3d ",y[i][j]); } }
textcolor(15);gotoxy(10,wherey()+2);
cprintf("Elementul Maximal din portiunea hasurata de sus este x[%d][%d] =
%d",rmax1,cmax1,y[rmax1][cmax1]);
gotoxy(10,wherey()+1);
cprintf("Elementul Maximal din portiunea hasurata de jos este x[%d][%d] =
%d",rmax2,cmax2,y[rmax2][cmax2]);
gotoxy(10,wherey()+2);
cprintf("Pentru iesire in meniu culegeti <ENTER>");
getch(); }
//calcularea sumei elementelor pare si inscrierea ei in fisier
void fisier (int y[50][50],int m) {
int i,j,k,s=0; FILE *f;
k=floor(float(m)/float(2));
//portiunea de sus a ariei hasurate
for(i=0;i<k;i++) {
for(j=i;j<(m-i);j++) {
if (fmod(y[i][j],2)==0) s+=y[i][j]; } }
//portiunea de jos a ariei hasurate.

```

```

for(i=k;i<m;i++) {
for(j=(m-i-1);j<=i;j++) {
if (fmod(y[i][j],2)==0) s+=y[i][j]; } }
if((f=fopen("c:/suma.txt", "w"))==NULL) {
clrscr();gotoxy(15,2);textcolor(15);
cprintf("Fisierul nu poate fi deschis"); goto exit;}
fprintf(f, "Suma elementelor pare din aria hasurata S=%d",s); fclose(f);
clrscr();gotoxy(15,2);textcolor(15);
cprintf("In fisierul c:\\suma.txt au fost inscise urmatoarele date:");
textcolor(11);gotoxy(15,wherey()+2);
cprintf("Suma elementelor pare din aria hasurata S=%d",s);
exit: textcolor(15);gotoxy(15,wherey()+2);
cprintf("Pentru iesire in meniu culegeti <ENTER>"); getch(); }
// aranjarea elementelor ariei hasurate in ordine crescatoare
void aranjare (int(*k)[50],int z) {
int i,j,g,c,min,aux,w;
w=floor(float(z)/float(2));
//aranjare in ordine crescatoare portiunea de sus
for(i=0;i<w;i++) {
for(j=i;j<(z-i);j++) {
min=*(k[i]+j);c=j;
for(g=j;g<(z-i);g++) {
if (*(k[i]+g)<min) {min=*(k[i]+g);c=g;}}
aux=*(k[i]+j); *(k[i]+j)=*(k[i]+c); *(k[i]+c)=aux;}}
//aranjare in ordine crescatoare portiunea de jos
for(i=w;i<z;i++) {
for(j=(z-i-1);j<=i;j++) {
min=*(k[i]+j);c=j;
for(g=j;g<=i;g++) {
if (*(k[i]+g)<min) {min=*(k[i]+g);c=g;}}
aux=*(k[i]+j); *(k[i]+j)=*(k[i]+c); *(k[i]+c)=aux;}}
clrscr();gotoxy(15,2);textcolor(15);
cprintf("Elementele fiecarui rind din aria hasurata");
textcolor(15);gotoxy(15,wherey()+1);
cprintf("au fost aranjate in ordine crescatoare");
textcolor(11);gotoxy(15,wherey()+2);
cprintf("Pentru a vedea rezultatul alegeți punctul 8 din meniu");
textcolor(15);gotoxy(15,wherey()+2);
cprintf("Pentru iesire in meniu culegeti <ENTER>"); getch(); }

```

```

//prelucrarea exceptiei in caz cind elementele masivului nu au valori
void gresala (int q) {
if (q==0) { clrscr();gotoxy(15,2);textcolor(12);
cprintf("Masivul nu poate fi prelucrat. Elementele nu au valori");
textcolor(15);gotoxy(15,wherey()+2);
cprintf("Pentru iesire in meniu culegeti <ENTER>"); getch(); } }
//programul principal
void main(void) {
int x[50][50]={0},t[50][50]={0};
int n,i,j,z=0,h=0; char w,v;
//afisarea meniului
m0:clrscr(); textcolor(15); gotoxy(15,2);
cprintf("Alege din meniu:\n"); gotoxy(15,4);
cprintf("1: Afisarea conditiilor initiale\n"); gotoxy(15,5);
cprintf("2: Completarea masivului in mod manual\n"); gotoxy(15,6);
cprintf("3: Completarea masivului in mod automat\n"); gotoxy(15,7);
cprintf("4: Afisarea masivului initial\n"); gotoxy(15,8);
cprintf("5: Aflarea elementelor maxime din portiunile hasurate\n"); gotoxy(15,9);
cprintf("6: Inscrierea sumei in fisier\n"); gotoxy(15,10);
cprintf("7: Aranjarea elementelor ariei hasurate\n"); gotoxy(15,11);
cprintf("8: Afisarea masivului final\n"); gotoxy(15,12);
cprintf("9: Iesire\n"); gotoxy(15,13);
// alegerea din meniu
w=getch();
switch (w) {
case '1': goto m1;
case '2': goto m2;
case '3': goto m3;
case '4': goto m4;
case '5': goto m5;
case '6': goto m6;
case '7': goto m7;
case '8': goto m8;
case '9': goto m9;
default : goto m0;}
m1: conditii(); goto m0;
m2: //completare manuala a masivului
clrscr(); z=1; gotoxy(15,2); textcolor(15);
cprintf("Culege marimea masivului x N=");

```

```

scanf("%d",&n); gotoxy(15,3);
printf("Culege elementele masivului x[%d][%d]\n",n,n);
for(i=0;i<n;i++) {
for(j=0;j<n;j++) {
gotoxy(15+(j*4),5+i);
scanf("%3d",&x[i][j]); } }
textcolor(15);gotoxy(10,wherey()+1);
printf("Pentru iesire in meniu culegeti <ENTER>"); getch(); goto m0;
m3: //completare automata a masivului
clrscr(); z=1; gotoxy(15,2); textcolor(15);
printf("Culege marimea masivului x N=");
scanf("%d",&n); randomize();
for(i=0;i<n;i++) {
for(j=0;j<n;j++) {
x[i][j]=random(100)-50; }}
// afisarea masivului initial
gotoxy(15,4); textcolor(15);
printf("Masivul initial x[%d][%d] este;\n",n,n);
for(i=0;i<n;i++) {
for(j=0;j<n;j++) {
textcolor(15); gotoxy(15+(j*4),6+i);
printf("%3d ",x[i][j]); } }
gotoxy(10,wherey()+2);
printf("Pentru iesire in meniu culegeti <ENTER>");
getch(); goto m0;
m4: if (z==1) {afisare(x,n); goto m0;};
    gresala(z); goto m0;
m5: if (z==1) {maxim(x,n); goto m0;};
    gresala(z); goto m0;
m6: if (z==1) {fisier(x,n); goto m0;};
    gresala(z); goto m0;
m7: if (z==1) { for(i=0;i<n;i++) for(j=0;j<n;j++) t[i][j]=x[i][j];
    aranjare(t,n); h=1; goto m0; };
    gresala(z); goto m0;
m8: if (h==1) {afisare(t,n); goto m0;};
    if (h==0) { clrscr();gotoxy(15,2);textcolor(12);
    printf("Masivul nu poate fi afisat. Elementele nu au fost aranjate");
    textcolor(15);gotoxy(15,wherey()+2);
    printf("Pentru iesire in meniu culegeti <ENTER>");

```

```
    getch(); goto m0;}  
m9: clrscr();  
    gotoxy(15,2); textcolor(15); cprintf("Intradevar doriti sa iesiti ? y/n");  
    v=getch(); if ((v=='n')||(v=='N')) goto m0;  
    gotoxy(15,4); textcolor(15);  
    cprintf("Pentru iesire din program culegeti <ENTER>"); getch(); }
```