

Laborator 2 Pointeri la tablouri

În acest capitol sunt prezentate considerații teoretice privind definirea și utilizarea pointerilor, operațiile permise și nepermise utilizând pointeri și câteva probleme rezolvate cu pointeri la date de tip standard: întreg, caracter și respectiv șir de caractere.

CONSIDERAȚII TEORETICE

a) Pointeri la tablouri unidimensionale (șiruri)

Accesul la elementele tablourilor se realizează în C++ în 2 moduri:

- prin intermediul indicilor
- prin aritmetica pointerilor. Avantaj: accesul e mult mai rapid și crește viteza de execuție a programului

Numele unui tablou (șir/matrice) - scris fără indice = **pointer constant** și este de tipul elementelor tabloului și are ca și valoare adresa primului element al tabloului

Se consideră tabloul unidimensional $a[i]$, $i=0, n-1$. Prin convenție: denumirea tabloului = pointerul la tablou = adresa primului element al tabloului ($a[0]$)

Notațiile de mai jos sunt echivalente:

a	$\&a$	$\&a[0]$
-----	-------	----------

unde a = constantă (pointer constant) și deci operațiile aritmetice cu pointeri nu se pot aplica asupra pointerilor constanți.

Din această cauză instrucțiunea: $a++$ este incorectă pentru că a este un pointer constant! De aceea pointerul la un tablou trebuie declarat ca și pointer variabil:

```
tip  $a[10]$ , *p;  
 $p=\&a[0]$ ; // sau  $p=a$ ; sau  $p=\&a$ ;
```

În acest caz $*p$ este o variabilă pointer, iar a este pointer constant.

De asemenea dacă se consideră tabloul unidimensional $a[i]$, $i=0, n-1$, a este pointerul constant la șir, iar notațiile de mai jos sunt echivalente:

termenul $a[i]$:	$*(a+i)$
adresa termenului $a[i]$: $\&a[i]$	$a+i$

Dacă se declară o variabilă de tip pointer la acest șir, $*p$ și se inițializează cu instrucțiunea $p=\&a[0]$; sau $p=a$; sau $p=\&a$, atunci notațiile de mai jos sunt echivalente:

$a[i]$	$*(a+i)$	$*(p+i)$	$p[i]$
--------	----------	----------	--------

b) Pointeri la tablouri bidimensionale (matrici)

Se consideră tabloul bidimensional $a[i][j]$, $i=0, m-1$; $j=0, n-1$. Prin convenție: denumirea tabloului = pointerul la tablou = adresa primului element al tabloului ($a[0][0]$).

Notațiile de mai jos sunt echivalente:

a	$\&a$	$\&a[0][0]$
-----	-------	-------------

unde a = numele tabloului= constantă (pointer constant) și deci operațiile aritmetice cu pointeri nu sunt permise

De aceea pointerul la tablou trebuie declarat ca și pointer variabil:

```
tip a[linii][coloane], *pa;
pa=&a[0][0]; // sau pa=a;
```

In acest caz *pa este o variabila de tip pointer.

Adresa lui a[i][j] se poate determina utilizând una din **notațiile echivalente** de mai jos:

&a[i][j] a[i]+j p+k *(a+i)+j

Valoarea lui a[i][j] se poate determina utilizând una din notațiile echivalente :

a[i][j] *(a[i]+j) *(p+k) (*(a+i)+j)

unde k=i*n+j .

PROBLEME REZOLVATE

Ex 1. Programul citește elementele a 2 vectori x[i] și y[i], de dimensiune egală cu n, și calculează și afișează produsul lor scalar, utilizând variabile de tip pointer.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> int main() {int i,n,x[100],y[100],s, *p,*q; //citire dimensiune vectori printf("Introduceti dimensiunea vectorilor x si y, n="); scanf("%d", &n); //citire elemente vectori x, y p=x,q=y; for(i=0;i<n;i++) { printf("x[%d]=",i); scanf("%d", p+i); printf("y[%d]=",i); scanf("%d", q+i);} //calcul prod scalar vectori s=0; for (i=0;i<n;i++,p++,q++) s+=*p* *q; printf(" produsul scalar (x,y)=%d\n", s); return 0;}</pre>	<pre>#include <iostream> using namespace std; int main() {int i,n,x[100],y[100],s, *p,*q; //citire dimensiune vectori cout<<"Introduceti dimensiunea vectorilor x si y, n="<<endl; cin>>n; //citire elemente vectori x, y p=x,q=y; for(i=0;i<n;i++) { cout<<"x["<<i<<"]="; cin>>*(p+i); cout<<"y["<<i<<"]="; cin>>*(q+i);} //calcul prod scalar vectori s=0; for (i=0;i<n;i++,p++,q++) s+=*p* *q; cout<<" produsul scalar (x,y)="<<s<<endl; return 0;}</pre>

Rezultate:

```
Introduceti dimensiunea vectorilor x si y, n=3
x[0]=1
y[0]=2
x[1]=1
y[1]=2
x[2]=1
y[2]=2
produsul scalar (x,y)=6
```

Aplicație:

Să se modifice programul de mai sus astfel încât, vectorii să aibă elemente de tip real (double)

Ex.2. Programul realizează inițializarea unui șir de numere întregi cu valori cuprinse în intervalele (1,10),(11,20), ... etc.,utilizând pointeri și aritmetica de pointeri.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> int main() {int i,j,a[100], *p; printf ("na.Initializare sir cu nr de la 1-10"); printf("\nfolosind numele sirului si un index\n"); for (i=1;i<11;i++) {a[i]=i; printf("%4d", a[i]);} printf ("\n\nb.Initializare sir cu nr. de la 11-20"); printf("\nfolosind un pointer si un index\n"); for (p=a,i=11;i<21;i++) {p[i]=i; printf("%4d", p[i]);} printf("\n\nc.Initializare sir cu nr. de la 21 la 30"); printf("\nfolosind un pointer si aritmetica pointerilor\n"); for (p=a,j=21;p<a+10;j++,p++) {*p=j;printf("%4d",*p);} printf ("\n\nd.Initializare sir cu nr. de la 31 la 40"); printf("\nfolosind un pointer si aritmetica de pointeri\n"); for (p=a,j=31;p<&a[10];j++,p++) {*p=j;printf("%4d",*p);} printf("\n"); return 0;}</pre>	<pre>#include <iostream> using namespace std; int main() {int i,j,a[100], *p; cout<<endl<<"a.Initializare sir cu nr de la 1- 10"<<endl; cout<<"folosind numele sirului si un index"<<endl; for (i=1;i<11;i++) {a[i]=i; cout<<a[i]<<" ";} cout<<"\n\nb.Initializare sir cu nr. de la 11- 20"<<endl; cout<<"folosind un pointer si un index"<<endl; for (p=a,i=11;i<21;i++) {p[i]=i; cout<< p[i]<<" ";} cout<<"\n\nc.Initializare sir cu nr. de la 21 la 30"<<endl; cout<<"folosind un pointer si aritmetica pointerilor"<<endl; for (p=a,j=21;p<a+10;j++,p++) {*p=j;cout<<*p<<" ";} cout<<"\n\nd.Initializare sir cu nr. de la 31 la 40"<<endl; cout<<"folosind un pointer si aritmetica de pointeri"<<endl; for (p=a,j=31;p<&a[10];j++,p++) {*p=j;cout<<*p<<" ";} cout<<endl; return 0;}</pre>

Rezultate:

```
a.Initializare sir cu nr de la 1-10
folosind numele sirului si un index
 1  2  3  4  5  6  7  8  9  10

b.Initializare sir cu nr. de la 11-20
folosind un pointer si un index
 11 12 13 14 15 16 17 18 19 20

c.Initializare sir cu nr. de la 21 la 30
folosind un pointer si aritmetica pointerilor
 21 22 23 24 25 26 27 28 29 30

d.Initializare sir cu nr. de la 31 la 40
folosind un pointer si aritmetica de pointeri
 31 32 33 34 35 36 37 38 39 40
```

Aplicație:

Să se modifice programul astfel încât să se afișeze șirul inițializat cu valori reprezentând cubul numerelor întregi din intervalul [1,100].

Ex. 3. Programul calculează suma tuturor elementelor și produsul elementelor strict pozitive ale unui șir de numere întregi, utilizând pointeri .

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> int main()</pre>	<pre>#include <iostream> using namespace std; int main()</pre>

<pre> {int a[10]={-1,0,1,10,2,-2,0,-5,1,-4}; int *pa, s=0, p=1,i; pa=a; printf("Tiparirea valorilor sirului\n a[i]\t *(a+i)\n"); for (i=0;i<10;i++) { printf("a[%d]=%3d, *(a+%d)=%3d\n",i,a[i],*(a+i)); s+=*(pa+i); //sau s+=a[i]; if(*(pa+i)>0) p*=*(pa+i); //sau if (a[i]>0) p*=a[i]; } printf("Suma tuturor elementelor sirului:\n"); printf("s=s+a[i]=%d\n",s); printf("Produsul elementelor strict pozitive:\n"); printf("p=p*a[i]=%d\n", p); return 0;} </pre>	<pre> {int a[10]={-1,0,1,10,2,-2,0,-5,1,-4}; int *pa, s=0, p=1,i; pa=a; cout<<"Tiparirea valorilor sirului\n a[i]\t *(a+i)"<<endl; for (i=0;i<10;i++) { cout<<"a["<<i<<"]="<<a[i]<<" , *(a+"<<i<<")=<<*(a+i)<<endl; s+=*(pa+i); //sau s+=a[i]; if(*(pa+i)>0) p*=*(pa+i);//sau if (a[i]>0) p*=a[i]; } cout<<"Suma tuturor elementelor sirului:"<<endl; cout<<"s=s+a[i]="<<s<<endl; cout<<"Produsul elementelor strict pozitive:"<<endl; cout<<"p=p*a[i]="<<p<<endl; return 0;} </pre>
---	---

Rezultate:

```

Tiparirea valorilor sirului
a[i] *(a+i)
a[0]= -1, *(a+0)= -1
a[1]= 0, *(a+1)= 0
a[2]= 1, *(a+2)= 1
a[3]= 10, *(a+3)= 10
a[4]= 2, *(a+4)= 2
a[5]= -2, *(a+5)= -2
a[6]= 0, *(a+6)= 0
a[7]= -5, *(a+7)= -5
a[8]= 1, *(a+8)= 1
a[9]= -4, *(a+9)= -4
Suma tuturor elementelor sirului:
s=s+a[i]=2
Produsul elementelor strict pozitive:
p=p*a[i]=20

```

Aplicație:

Să se modifice programul astfel încât să se citească șirul de la tastatură (citind înainte dimensiunea sa) și să se calculeze și afișeze suma elementelor din intervalul [1,100] și produsul elementelor din intervalul [10,50].

Ex.4. Programul afișează un șir de caractere in 2 moduri: fără pointer și utilizând un pointer indexat.

Varianta in C	Varianta in C++
<pre> #include <stdio.h> #include <stdlib.h> int main() {char sir[]="Pointeri la siruri de caractere", *p; int i; p=sir; printf("Sirul tiparit fara pointer:\n"); printf("%s",sir); printf("\n\nSirul tiparit cu pointer indexat:\n"); for (i=0;p[i];i++) printf("%c", p[i]); return 0;} </pre>	<pre> #include <iostream> using namespace std; int main() {char sir[]="Pointeri la siruri de caractere", *p; int i; p=sir; cout<<"Sirul tiparit fara pointer:"<<endl; cout<< sir<<endl<<endl; cout<<"Sirul tiparit cu pointer indexat:"<<endl; for (i=0;p[i];i++) cout<<p[i]; return 0;} </pre>

Rezultate:

Aplicație:

Sirul tiparit fara pointer:
Pointeri la siruri de caractere

Sirul tiparit cu pointer indexat:
Pointeri la siruri de caractere

Să se modifice programul astfel încât să se afișeze șirul inversat, prin pointeri.

Ex.5. Programul citește un șir de numere reale $a[i]$, $i=1,n$, n întreg citit de la tastatură, și creează un șir $b[i]$ format după formula: $b_i = \frac{a_i}{(1 - a_i)^2}$, utilizând pointeri.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include<stdlib.h> #include <math.h> int main() {double a[100],*p; int i, n; p=a; printf("n=");scanf("%d", &n); for(i=0;i<n;i++) { printf("a[%d]=",i); scanf("%lf", p+i);} for (i=0;i<n;i++) {if (pow((1-*(p+i)),2)==0) {printf("numitor 0!"); exit(1); } else printf("b[%d]=%.2lf\n",i,*(p+i)/pow((1-*(p+i)),2));} return 0;}</pre>	<pre>#include <iostream> #include <math.h> #include<stdlib.h> using namespace std; int main() {double a[100],*p; int i, n; p=a; cout<<"n=";cin>>n; for(i=0;i<n;i++) { cout<<"a["<<i<<"]="; cin>>*(p+i);} for (i=0;i<n;i++) {if (pow((1-*(p+i)),2)==0) {cout<<"numitor 0!"; exit(1); } else cout<<"b["<<i<<"]="<<*(p+i)/pow((1- *(p+i)),2)<<endl;}</pre>

Rezultate:

```
n=5
a[0]=10
a[1]=11
a[2]=12
a[3]=13
a[4]=14
b[0]=0.12
b[1]=0.11
b[2]=0.10
b[3]=0.09
b[4]=0.08
```

Aplicație:

Să se modifice programul astfel încât să se afișeze șirul $b_i = 5a_i + 7$ prin pointeri.

Ex. 6. Programul calculează și afișează primii 10 de termeni ai celor 2 șiruri definite după formulele de mai jos:

$$a_n = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \dots - \frac{1}{2n}, \quad b_n = \frac{1}{n+1} + \frac{1}{n+2} + \frac{1}{n+3} + \dots + \frac{1}{2n}, \text{ utilizând pointeri.}$$

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> # include<math.h> int main() { float a[100],b[100],*pa,*pb; int i,n; pa=a;pb=b; printf(" n=");scanf("%d",&n); *(pa+1)=1;printf("a[1]=%lf", *(pa+1)); *(pb+1)=1./(n+1); printf("\tb[1]=%lf", *(pb+1));</pre>	<pre>#include <iostream> # include<math.h> using namespace std; int main() { float a[100],b[100],*pa,*pb; int i,n; pa=a;pb=b; *(pa+1)=1;cout<<"a[1]="<<*(pa+1); *(pb+1)=1./(n+1); cout<<"\tb[1]="<<*(pb+1); for(i=2;i<=n;i++)</pre>

<pre>for(i=2;i<=n;i++) { *(pa+i)=*(pa+i-1)+pow(-1,i+1)*(1./i); printf("\na[%d]=%f",i,*(pa+i)); *(pb+i)=*(pb+i-1)+1./(n+i); printf("\tb[%d]=%f",i,*(pb+i));}return 0;}</pre>	<pre>{ *(pa+i)=*(pa+i-1)+pow(-1,i+1)*(1./i); cout<<"\na["<<i<<"]="<<*(pa+i); *(pb+i)=*(pb+i-1)+1./(n+i); cout<<"\tb["<<i<<"]="<<*(pb+i);} return 0;}</pre>
--	--

Rezultate:

```
n=10
a[1]=1.000000 b[1]=0.090909
a[2]=0.500000 b[2]=0.174242
a[3]=0.833333 b[3]=0.251166
a[4]=0.583333 b[4]=0.322594
a[5]=0.783333 b[5]=0.389261
a[6]=0.616667 b[6]=0.451761
a[7]=0.759524 b[7]=0.510584
a[8]=0.634524 b[8]=0.566140
a[9]=0.745635 b[9]=0.618771
a[10]=0.645635 b[10]=0.668771
```

Aplicație:

Să se modifice programul astfel încât să se afișeze șirul format din produsul elementelor $a[i]*b[i]$ utilizând pointeri.

Ex.7. Programul citește elementele unei matrici de dimensiune $n \times m$, și le afișează prin trei metode diferite utilizând pointeri.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> int main() {int i, j, n,m,a[100][100]; printf ("n=");scanf("%d", &n); printf ("m=");scanf("%d", &m); for (i=0;i<n;i++) for (j=0;j<m;j++) {printf(" a[%d,%d]=", i,j); scanf("%d", *(a+i)+j); } for (i=0;i<n;i++) for (j=0;j<m;j++) {printf("\na[%d][%d]=%3d", i,j,a[i][j]); printf(" *a[%d]+%d=%3d", i,j,*(a[i]+j)); printf(" *(a+%d)+%2d=%d", i,j,*(a+i+j));} printf("\n"); return 0;}</pre>	<pre>#include <iostream> using namespace std; int main() {int i, j, n,m,a[100][100]; cout<<" n=";cin>>n; cout<<" m=";cin>>m; for (i=0;i<n;i++) for (j=0;j<m;j++) {cout<<" a["<<i<<","<<j<<"]="<<*(a+i)+j); } for (i=0;i<n;i++) for (j=0;j<m;j++) {cout<<"\na["<<i<<"]["<<j<<"]="<<a[i][j]; cout<<" *a["<<i<<"+"<<j<<"]="<<*(a[i]+j); cout<<" *(a+"<<i<<"+"<<j<<"]="<<*(a+i+j);} cout<<endl;return 0;}</pre>

Rezultate:

```
n=2
m=3
a[0,0]=1
a[0,1]=2
a[0,2]=3
a[1,0]=4
a[1,1]=5
a[1,2]=6

a[0][0]= 1 *a[0]+0= 1 *(a+0)+ 0)=1
a[0][1]= 2 *a[0]+1= 2 *(a+0)+ 1)=2
a[0][2]= 3 *a[0]+2= 3 *(a+0)+ 2)=3
a[1][0]= 4 *a[1]+0= 4 *(a+1)+ 0)=4
a[1][1]= 5 *a[1]+1= 5 *(a+1)+ 1)=5
a[1][2]= 6 *a[1]+2= 6 *(a+1)+ 2)=6
```

Aplicație:

Să se modifice programul astfel încât să se calculeze și afișeze elementele matricii multiplicat cu 100.

Ex.8. Programul determină și afișează maximul elementelor unei matrici de nxm elemente citite de la tastatură utilizând pointeri.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> int main() {int max; int a[10][10],i,j,n,m,*p; p=&a[0][0]; printf("n=");scanf("%d", &n); printf("m=");scanf("%d", &m); for (i=0;i<n;i++) for (j=0;j<m;j++) { printf("a[%d][%d]=",i,j); scanf("%d", *(a+i)+j);} max=*p; for (i=0; i < n; i++) for (j=0; j < m; j++) if (max < (*(a+i)+j)) max = (*(a+i)+j); printf("\n max din matrice este %d\n",max); return 0;}</pre>	<pre>#include <iostream> using namespace std; int main() {int maxi; int a[10][10],i,j,n,m,*p; p=&a[0][0]; cout<<"n=";cin>>n; cout<<"m=";cin>>m; for (i=0;i<n;i++) for (j=0;j<m;j++) { cout<<"a["<<i<<"]["<<j<<"]="; cin>>* (a+i)+j);} maxi=*p; for (i=0; i < n; i++) for (j=0; j < m; j++) if (maxi < (*(a+i)+j)) maxi = (*(a+i)+j); cout<<endl<<"max din matrice este "<<maxi<<endl; return 0;}</pre>

Rezultate:

```
n=2
m=2
a[0][0]=1
a[0][1]=2
a[1][0]=3
a[1][1]=4

max din matrice este 4
```

Aplicație:

Să se modifice programul astfel încât să se determine și minimul elementelor matricii și poziția minimului și a maximului în matrice.

Ex. 9. Programul definește un pointer la un sir de caractere prin intermediul caruia se va afișa sirul inversat.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <string.h> int main() {int i; char c[100], *p; printf("Introduceti un text oarecare:\n");gets(c); p=c; //<=>p=&c, sau p=&c[0] printf("Ati introdus textul:\n");printf(p); printf("\nTextul inversat este:\n"); for (i=strlen(p)-1;i>=0;i--) printf("%c", p[i]); printf("\n"); return 0;}</pre>	<pre>#include <iostream> #include <string.h> using namespace std; int main() {int i; char c[100], *p; cout<<"Introduceti un text oarecare:"<<endl; cin.get(c,100); p=c; //<=>p=&c, sau p=&c[0] cout<<"Ati introdus textul:"<<endl<<p; cout<<endl<<"Textul inversat este:"<<endl; for (i=strlen(p)-1;i>=0;i--) cout<<p[i]; cout<<endl; return 0;}</pre>

Rezultate:

Aplicatie:

```

Introduceti un text oarecare:
Programare in C/C++
Ati introdus textul:
Programare in C/C++
Textul inversat este:
+C/C ni eramargorP

```

Sa se modifice programul astfel încât sa se afișeze sirul inversat, utilizand pointeri si o alta metoda de tiparire

Ex.10. Programul afisează un sir de caractere (caracter cu caracter) introdus de la tastatura prin intermediul pointerilor si numără cate caractere are acest sir.

Varianta I in C	Varianta II in C++
<pre> #include <stdio.h> #include <string.h> int main() {int i=0; char c[100], *p; printf("Introduceti un text oarecare:\n");gets(c); p=c; //<=>p=&c, sau p=&c[0] printf("Ati introdus textul:\n"); while(*p) {i++;putchar(*p++);} printf("\nNr de caractere (inclusiv spatiile) din text:%d\n", i); return 0; } </pre>	<pre> #include <iostream> #include <stdio.h> #include <string.h> using namespace std; int main() {int i=0; char c[100], *p; cout<<"Introduceti un text oarecare:"<<endl; cin.get(c,100); p=c; //<=>p=&c, sau p=&c[0] cout<<"Ati introdus textul:"<<endl; while(*p) {i++;putchar(*p++);} cout<<endl<<"Nr de caractere (inclusiv spatiile) din text:"<< i; return 0;} </pre>
Varianta II in C	Varianta II in C++
<pre> #include <stdio.h> #include <string.h> int main() {int i=0; char c[100], *p; printf("Introduceti un text oarecare:\n");gets(c); p=c; //<=>p=&c, sau p=&c[0] printf("Ati introdus textul:\n"); for (i=0;p[i];i++) printf("%c", p[i]); printf("\nNr de caractere (inclusiv spatiile) din text:%d\n", i); return 0;} </pre>	<pre> #include <iostream> #include <stdio.h> #include <string.h> using namespace std; int main() {int i=0; char c[100], *p; cout<<"Introduceti un text oarecare:"<<endl; cin.get(c,100); p=c; //<=>p=&c, sau p=&c[0] cout<<"Ati introdus textul:"<<endl; for (i=0;p[i];i++) cout<<p[i]; cout<<endl<<"Nr de caractere (inclusiv spatiile) din text:"<< i; return 0;} </pre>

Rezultate:

```

Introduceti un text oarecare:
Programare in C/C++
Ati introdus textul:
Programare in C/C++
Nr de caractere (inclusiv spatiile) din text:19

```

Aplicatie:

Să se modifice programul astfel incat sa se utilizeze o altă modalitate de afișare a șirului.

Ex.12. Programul realizeaza copierea unui sir in alt sir utilizand 2 pointeri la siruri si 3 functii (functiile de citire/afisare sir si functia de copiere prin intermediul acestor pointeri).Functia de copiere este similara cu functia strcpy din biblioteca <string.h>

Varianta in C	Varianta in C++
<pre>#include <stdio.h> void strcpy(char *s,char *t); //copiază t în s void scriesir(char *s); void citiresir(char *s); int main () {char sir1[100],sir2[100]; char *s,*t; s=sir1; t=sir2; printf("Introduceti primul sir: "); citiresir(s);scriesir(s); printf("\n\nIntroduceti sirul al doilea: "); citiresir(t);scriesir(t); strcpy(s,t); printf("\nCopiem sirul al doilea peste primul sir\n"); printf("Dupa copiere sirurile sunt :\n"); printf("primul sir: ");scriesir(s);printf("\n"); printf("sirul al doilea: "); scriesir(t);printf("\n"); return 0;} void citiresir(char *p) { gets(p);} void scriesir(char *q) { while(*q) putchar(*q++);} void strcpy(char *p,char *q) {while ((*p=*q)!='\0') {p++; q++;}}</pre>	<pre>#include <iostream> #include <stdio.h> using namespace std; void strcpy(char *s,char *t); //copiază t în s void scriesir(char *s); void citiresir(char *s); int main () {char sir1[100],sir2[100]; char *s,*t; s=sir1; t=sir2; cout<<"Introduceti primul sir: "; citiresir(s);scriesir(s); cout<<endl<<endl<<"Introduceti sirul al doilea: "; citiresir(t);scriesir(t); strcpy(s,t); cout<<endl<<"Copiem sirul al doilea peste primul sir"<<endl; cout<<"Dupa copiere sirurile sunt :"<<endl; cout<<"primul sir: ";scriesir(s);cout<<endl; cout<<"sirul al doilea: "; scriesir(t);cout<<endl; return 0;} void citiresir(char *p) { gets(p);} void scriesir(char *q) { while(*q) putchar(*q++);} void strcpy(char *p,char *q) {while ((*p=*q)!='\0') {p++; q++;}}</pre>

Rezultate:

```
Introduceti primul sir: Curs Programare
Curs Programare
Introduceti sirul al doilea: C/C++
C/C++
Copiem sirul al doilea peste primul sir
Dupa copiere sirurile sunt :
primul sir: C/C++
sirul al doilea: C/C++
```

Aplicatie:

In mod similar, sa se scrie programul care rezolva concatenarea și compararea a 2 siruri utilizând pointeri la șiruri de caractere.

Ex.13. Programul afiseaza 2 siruri de caractere prin intermediul pointerilor în diverse variante și schimba sirurile între ele.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <string.h> int main() {char *p="Curs de "; char *q="Programare in C/C++!"; char *aux; int i; printf("Afisarea sirurilor de caractere\n"); printf("prin intermediul pointerilor\n"); //varianta 1 de afisare a sirurilor printf("\nVarianta I\n"); printf("pointerul p indica:"); printf(p); printf("\n");</pre>	<pre>#include <iostream> #include <stdio.h> #include <string.h> using namespace std; int main() {char *p="Curs de "; char *q="Programare in C/C++!"; char *aux; int i; cout<<"Afisarea sirurilor de caractere"<<endl; cout<<"prin intermediul pointerilor"<<endl; //varianta 1 de afisare a sirurilor</pre>

```

printf("pointerul q indica:"); printf(q); printf("\n");
//varianta 2 de afisare a sirurilor
printf("\n\nVarianta II\n");
printf("pointerul p indica:");
for (i=0; i<strlen(p)+1;i++) printf("%c",
p[i]);printf("\n");
printf("pointerul q indica:");
for (i=0; i<strlen(q)+1;i++) printf("%c", q[i]);
printf("\n");
//varianta 3 de afisare a sirurilor
printf("\n\nVarianta III\n");
printf("pointerul p indica:");printf("%s\n", p);
printf("pointerul q indica:");printf("%s\n", q);
printf("pointerul p indica:");
//inversarea sirurilor
printf("\n\ninversam sirurile prin inversarea
pointerilor\n");
aux=p; p=q; q=aux;
printf("pointerul p indica:"); printf(p); printf("\n");
printf("pointerul q indica:"); printf(q); printf("\n");
return 0;}

```

```

cout<<"Varianta I"<<endl;
cout<<"pointerul p indica:"<<p<<endl;
cout<<"pointerul q indica:"<<q<<endl;
//varianta 2 de afisare a sirurilor
cout<<endl<<"Varianta II"<<endl;
cout<<"pointerul p indica:"<<endl;
for (i=0; i<strlen(p)+1;i++) cout<<p[i]<<endl;
cout<<"pointerul q indica:";
for (i=0; i<strlen(q)+1;i++) cout<<p[i]<<endl;
//varianta 3 de afisare a sirurilor
cout<<endl<<"Varianta III"<<endl;
cout<<"pointerul p indica:"<<p<<endl;
cout<<"pointerul q indica:"<< q<<endl;
cout<<"pointerul p indica:"<<endl;
//inversarea sirurilor
cout<<"inversam sirurile prin inversarea
pointerilor"<<endl;
aux=p; p=q; q=aux;
cout<<"pointerul p indica:"<<p<<endl;
cout<<"pointerul q indica:"<<q<<endl;
return 0;}

```

Rezultate:

Aplicatie:

```

Afisarea sirurilor de caractere
prin intermediul pointerilor

Varianta I
pointerul p indica:Curs de
pointerul q indica:Programare in C/C++!

Varianta II
pointerul p indica:Curs de
pointerul q indica:Programare in C/C++!

Varianta III
pointerul p indica:Curs de
pointerul q indica:Programare in C/C++!
pointerul p indica:

inversam sirurile prin inversarea pointerilor
pointerul p indica:Programare in C/C++!
pointerul q indica:Curs de

```

Sa se modifice programul astfel încât sa se concateneze sirurile utilizand pointeri .

PROBLEME PROPUSE

1. Să se scrie programul care ordonează un șir de numere reale prin cel puțin două metode diferite de sortare, utilizând pointeri.
2. Să se scrie un program care calculează primii 30 de termeni ai celor 2 șiruri definite după formulele de mai jos:

$$a_n = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \dots - \frac{1}{2n} \quad b_n = \frac{1}{n+1} + \frac{1}{n+2} + \frac{1}{n+3} + \dots + \frac{1}{2n} \text{ utilizând pointeri.}$$

3. Sa se scrie un program care realizează citirea și afișarea elementelor unei matrici diagonale, a unei matrici triunghiulare superioare și a unei matrici tridiagonale, utilizând pointeri.

4. Să se scrie un program care determină dacă o matrice pătratică este pătrat magic, utilizând pointeri. Dacă se consideră că dimensiunea matricii este $(n \times n)$, n impar, și matricea conține toate numerele naturale de la 1 la n^2 astfel încât suma elementelor de pe fiecare linie, coloana și diagonală este aceeași .

Ex.

6 1 8

7 5 3

2 9 4

5. Sa se scrie un program care citeste de la tastatura un sir $x[i]$, $i=1, n$ (n intreg) numere reale și calculeaza si afiseaza valoarea functiei : $f(x) = \begin{cases} x^2 - 1, & x > 0 \\ -5, & x < 0 \end{cases}$, utilizand pointeri la sir