



Facultatea de Inginerie Electrică



UNIVERSITATEA TEHNICĂ  
DIN CLUJ-NAPOCA



# PCLP 2

## Programarea calculatoarelor si limbaje de programare 2

PCLP2

An I semestrul II



*"Coding is easy when you C it in action."*

## Cap. 4

# Structuri in C/C++

---

4.1 Clasificare. Definitie. Exemple

4.2 Accesul la elemente

4.3 Initializarea elementelor

4.4 Atribuire in structuri

4.5 Pointeri la structuri

4.6 Tablouri ca elemente de structuri

4.7 Structuri imbricate

4.8 Tablouri de structuri

4.9 Transmiterea structurilor catre functii

4.10 Alocarea dinamica a memoriei pentru structuri

# 4. Structuri in C/C++

## 4.1 Clasificare. Definare. Exemple

**O variabila tablou poate contine elemente de tipuri diferite?**

**R: NU.**

**Structura de date:** o colectie de variabile (de tipuri diferite) referite sub un singur nume.

**Clasificare tipuri de date structurate in C**

- structuri
- uniuni
- enumerari
- typedef
- campuri de biti

# 4. Structuri in C/C++

## 4.1 Clasificare. Definitie. Exemple

### DEFINIRE

**Structura:** colectie de variabile ce pot fi de **tipuri diferite** (spre deosebire de tablouri) reunite sub acelasi nume. La declararea unei structuri se alocă automat memorie pentru toti membri structurii

**Elementele (membri):** sunt variabilele din componenta unei structuri care pot fi de orice tip (standard, pointeri, siruri, alte structuri, etc) iar numele lor pot fi aceleasi cu elementele altei structuri din acelasi program

### SINTAXA

Declarare tip structura:

```
struct nume_tip  
{ tip var1;  
  tip var2;  
  ...  
};
```

Declarare variabile de tip structura:

```
struct nume_tip lista_variabile_struct;
```

Declarare tip si variabile intr-o singura instructiune:

```
struct nume_tip  
{ tip var1;  
  tip var2;  
  ...  
} lista_variabile_struct;
```

# 4. Structuri in C/C++

## 4.1 Clasificare. Definire. Exemple

### EXAMPLE

Ex.1. : Declararea tipului structura student, declararea variabilelor tip structura

```
struct student {           // declararea tipului structurii
    char nume[20];
    char prenume[20];
    double note[10];
    int an_studiu;
    int cod; };
struct student stud1,stud2,stud3; //declarare variabile de tip student
```

Ex.2. : Declararea unei structuri de tip student, intr-o singura instructiune

```
struct student {           // declararea tipului structurii
    char nume[20];
    char prenume[20];
    double note[10];
    int an_studiu;
    int cod; } stud1,stud2,stud3; //declarare variabile
```

# 4. Structuri in C/C++

## 4.1 Clasificare. Definire. Exemple

### EXEMPLE

Ex.3. : Declararea unei structuri de tip adresa:

```
struct adrese // adrese =numele tipului structurii
{ char localitate[30]; //elementele structurii
  char strada[40];
  char judet[20];
  int nr;
};
struct adrese adr; //declararea variabilei adr de tip adrese
```

Alocarea spatiului de memorie pentru variabila de tip structura numita adr: 94 octeti

localitate	30 octeti
strada	40 octeti
judet	20 octeti
nr	4 octeti

# 4. Structuri in C/C++

## 4.1 Clasificare. Definire. Exemple

### EXAMPLE

Ex.4. : Declararea unei structuri de tip coordonatele 2D ale unui punct:

```
struct punct2D          // punct2D =numele tipului structurii
{
    double x;
    double y;
};
struct punct2D A,B,C;  //declararea variabilelor A,B,C
```

Ex.5. : Declararea unei structuri de tip coordonatele 3D ale unui punct:

```
struct punct3D          // punct3D =numele tipului structurii
{
    double x;
    double y;
    double z;
};
struct punct3D A,B,C;  //declararea variabilelor A,B,C
```

# 4. Structuri in C/C++

## 4.2 Acces la elementele structurii

### DEFINIRE

Operatorul `.` permite accesul la elementele unei structuri : `nume_struct.nume_membru`

### EXEMPLE

Ex.1. : Fie structura de tip `data`. Inializare : cu valori pentru fiecare element

```
#include <stdio.h>
struct data {          //definitie globala de tip structura
    int ziua;
    int luna;
    int an; };

int main()
{struct data azi; //declarare variabila structura
azi.luna = 3;     //initializare elemente structura
azi.ziua = 30;
azi.an = 2022;
printf("Data de azi este %d/%d/%d.\n", azi.ziua,azi.luna, azi.an); return 0;}
```

Data de azi este 30/3/2022.



`nume_tip` si `lista_variabile_struct` sunt optionale, dar nu pot lipsi ambele simultan



# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### DEFINIRE

**Initializarea:** se realizeaza similar cu initializarea sirurilor, utilizand declararea de tip static a structurii, constantele de initializare fiind separate prin virgula

### EXEMPLE

Ex. 2: In structura de tip data se initializeaza elementele cu constante :

```
struct data {  
    int luna;  
    int ziua;  
    int an;  
};  
struct data azi = { 30,3,2022 }, maine={ 31,3,2022 };
```

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXEMPLE

Ex.3. : Fie structura de tip data. Initializare : prin enumerarea in acolada

```
#include <stdio.h>
struct data {          //definitie globala de tip
    int ziua;
    int luna;
    int an; };

int main()
{struct data azi={30,3,2024};
printf("Data de azi este %d/%d/%d.\n", azi.ziua,azi.luna, azi.an);
return 0;}
```

```
Data de azi este 30/3/2024.
```

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXEMPLE

Ex.4. : In structura de tip **adrese** atribuim valori unor elemente ale structurii:

```
#include <stdio.h>
int main()
{struct adrese
  { char localitate[30];
    char strada[40];
    int nr;
    char judet[20];
  } adr;
  adr.nr=30 ; //accesul la elementul cod al structurii
  printf("%d\n", adr.nr);
  gets(adr.strada); //accesul la elementul strada al structurii
  printf("Strada: %s, nr: %d", adr.strada, adr.nr);
  return 0;}
```

```
30
Baritiu
Strada: Baritiu, nr: 30
```

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXAMPLE

Ex. 5: In structura de tip **student** se initializeaza elementele cu constante. Atentie la ordinea de initializare; sirurile de caractere se introduc intre ghilimele

```
struct student {  
    char name[20];  
    int grupa;  
    float media;  
} std1 = { "Ispas Ion", 1113, 7.3 },  
  std2 = { "Popa Anca", 1111, 9.75};
```

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXAMPLE

Ex. 6: Preluarea datelor de la tastatura:

```
#include <stdio.h>
int main()
{ struct adrese { char nume[30];
                 char strada[40];
                 int nr, ap;
                 char local[20];
                 int cod; } adr;

printf("Numele si prenumele: "); gets(adr.nume);
printf("Strada:"); gets(adr.strada); adr.nr=11;
printf("Localitate:"); gets(adr.local);adr.cod=400200;//cod unic de strada
printf("apartamentul:"); scanf("%d", &adr.ap);
printf("\nAti introdus urmatoarele date:\n");
printf("%s\nAdresa:%s,%d,ap.%d\n",adr.nume,adr.strada,adr.nr,adr.ap);
printf("%s\nCod:%d\n", adr.local,adr.cod); return 0;}
```

```
Numele si prenumele: Popa Ioan
Strada:Horea
Localitate:Cluj-Napoca
apartamentul:24
```

```
Ati introdus urmatoarele date:
Popa Ioan
Adresa:Horea,11,ap.24
Cluj-Napoca
Cod:400200
```

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXEMPLE

Ex. 6: Să se scrie un program care ilustrează operațiile de bază specifice numerelor complexe, și afișează modulul și argumentele acestora utilizând o variabilă de tip structură cu doua câmpuri: partea reală și partea imaginară.

Operații:

- *Adunarea/scăderea:*  $x+iy = (x_1+iy_1) \pm (x_2+iy_2)$
- *Produsul:*  $x+iy = (x_1+iy_1)(x_2+iy_2) = (x_1x_2 - y_1y_2) + i(x_1y_2 + x_2y_1)$
- *Câtul:*  $x + iy = \frac{x_1 + iy_1}{x_2 + iy_2} = \frac{x_1x_2 + y_1y_2}{x_2^2 + y_2^2} + i \frac{x_2y_1 - x_1y_2}{x_2^2 + y_2^2}$ ,  
 $x_2^2 + y_2^2 > 0$
- *Valoarea absolută:*  $|x + iy| = \sqrt{x^2 + y^2}$ .
- *Argumentul:*  $arg(x + iy) = \begin{cases} \arctg(\frac{y}{x}), \arctg(\frac{y}{x}) \geq 0 \\ \arctg(\frac{y}{x}) + 2\pi, \arctg(\frac{y}{x}) < 0 \end{cases}$

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXAMPLE

Ex. 6: Să se scrie un program care ilustrează operațiile de bază specifice numerelor complexe

```
#include <stdio.h>
#include <math.h>
#define PI 3.1415926
int main()
{struct complex { double re; double im;} z,y;
//citirea nr.complexe
printf("Introduceti z=a+bj\n");
printf("a= "); scanf("%lf",&z.re); printf("b= "); scanf("%lf",&z.im);
printf("z=%2.2lf+%2.2lfj\n", z.re, z.im);
printf("Introduceti y=c+dj\n");printf("c= ");
scanf("%lf",&y.re); printf("d= "); scanf("%lf",&y.im);
printf("y=%2.2lf+%2.2lfj\n", y.re, y.im);
//adunarea nr. complexe
printf("adunarea nr. complexe:\n");
printf("(%2.2lf+%2.2lfj)+(%2.2lf+%2.2lfj)=%2.2f+%2.2fj\n",z.re,z.im,y.re,y.im,z.re+y.re,z.im+y.im );
// inmultirea a nr. complexe printf("inmultirea nr. complexe:\n");
printf("(%2.2lf+%2.2lfj)*(%2.2lf+%2.2lfj)=%2.2f+%2.2fj\n",z.re,z.im, y.re, y.im, z.re*y.re-z.im*y.im, z.re*y.im+z.im*y.re);
```

```
Introduceti z=a+bj
a= 2
b= 1
z=2.00+1.00j
Introduceti y=c+dj
c= 3
d= 4
y=3.00+4.00j
adunarea nr. complexe:
(2.00+1.00j)+(3.00+4.00j)=5.00+5.00j
2.00+1.00j)*(3.00+4.00j)=2.00+11.00j
inpartirea nr. complexe:
2.00+1.00j)/(3.00+4.00j)=
=0.82+-0.29j
modulul lui 2.00 + 1.00j = 2.24
modulul lui 3.00 + 4.00j = 5.00
arg(2.00+1.00 j)=0.46
arg(3.00+4.00 j)=0.93
```

continuare

# 4. Structuri in C/C++

## 4.3 Initializare elemente structura

### EXAMPLE

```
//impartirea nr. complexe
printf("inpartirea nr. complexe:\n");
if((z.im*z.im+y.im*y.im)>0)
{printf("%2.2lf+%2.2lfj)/(%2.2f+%2.2fj)= \n",z.re,z.im, y.re,y.im);
printf("=%2.2lf+%2.2lfj\n", (z.re*z.im+y.re*y.im)/(z.im*z.im+y.im*y.im), (z.im* y.re z.re*y.im)/
(z.im*z.im+y.im*y.im))}
else {printf("numarator =0\n") ;}
//modulul nr. complex
printf("modulul lui %2.2lf + %2.2lfj = %2.2f\n", z.re, z.im, sqrt(z.re*z.re +z.im*z.im));
printf("modulul lui %2.2lf + %2.2lfj = %2.2f\n", y.re, y.im, sqrt(y.re*y.re +y.im*y.im));
//argumentul nr.complex
if (atan2(z.im,z.re)>=0.) printf("arg(%2.2lf+%2.2lf j)=%2.2lf\n", z.re,z.im, atan2(z.im,z.re));
else printf("arg(%2.2lf+%2.2lfj)=%2.2lf\n", z.re,z.im, atan2(z.im,z.re)+2*PI);
if (atan2(y.im,y.re)>=0.) printf("arg(%2.2lf+%2.2lf j)=%2.2lf\n", y.re,y.im, atan2(y.im,y.re));
else printf("arg(%2.2lf+%2.2lfj)=%2.2lf\n", y.re,y.im, atan2(y.im,y.re) +2*PI );
return 0;}
```



# 4. Structuri in C/C++

## 4.4 Atribuirii in structuri

### DEFINIRE

**Atribuirea:** Datele continute intr-o structura poate fi atribuite altei structuri de acelasi tip printr-o singura instructiune de atribuire

### EXEMPLE

Ex.1. : Atribuim variabilei structura y ,structura x:

```
#include <stdio.h>
int main()
{struct
  { int a;
    int b;
  } x,y;
x.a=10; x.b=5;
y=x; //atribuie intreaga structura (element cu element) x lui y
printf("%d %d\n", y.a,y.b);
return 0;}
```

# 4. Structuri in C/C++

## 4.5 Pointeri la structuri

### DEFINIRE

Declararea unui pointer la o structura:

```
struct tip_struct *pointer
```

unde **pointer** va fi o variabila pointer catre date de tipul structurii tip\_struct

Accesul la elementele structurii prin pointer:

```
operatorul "sageata"-> si (*pointer).
```

### EXEMPLE

Ex.1. : accesarea elementelor structurii folosind pointeri :

```
struct tip {int i;  
            float bilant;  
            char nume[80];  
            } persoana;  
struct tip *p; p=&persoana; //declara un pointer la structura si-l initializeaza cu adresa ei  
p->bilant=0; // sau (*p).bilant=0; //persoana.bilant=0; //permite accesul la membrul bilant din  
            variabila persoana  
p->i++; //incrementeaza membrul i al structurii
```

# 4. Structuri in C/C++

## 4.5 Pointeri la structuri

### EXEMPLE

Ex.2. : accesarea elementelor structurii si suma a 2 nr. complexe folosind pointeri:

```
struct complex { double real;  
                 double imag; };
```

```
struct complex a,b,c ,*p, *q, *r;
```

```
p=&a; q=&b; r=&c; //p, q si r pointeri la variabilele structura a,b,c
```

1

```
a.real=0.5; //atribuirea de valori nr. complexe prin accesul direct la elemente  
b.imag=1.2;
```

...

2

```
(*p).real =0.5; //atribuirea de valori nr. complexe prin pointeri utilizand .  
(*q).imag=1.2;
```

...

3

```
p->real=0.5; //atribuirea de valori nr. complexe prin pointeri utilizand ->  
q->imag=1.2;
```

...

```
c.real=a.real+b.real; //sau (*r).real= (*p).real + (*q).real ;  
c.imag=a.imag+b.imag; // sau r->imag=p->imag+q->imag;
```

# 4. Structuri in C/C++

## 4.6 Tablouri ca elemente de structuri

### DEFINIRE

Declararea unor tablouri ca elemente ale unei structuri:

```
struct nume_tip_struct
{ tip var1;
  tip tab[max]; //sir
  tip mat[max][max]; //matrice
  ...
} struct1,struct2,...,structi,..., *p;
```

p=&structi ;

Accesul la elementul de tip tablou :

structi.tab[i]=expresie sau constanta;

structi.mat[i][j]=expresie sau constanta

p->tab[i]=expresie sau constanta;

p->mat[i][j]=expresie sau constanta

(\*p).tab[i]=expresie sau constanta;

(\*p).tab[i][j]=expresie sau constanta;

# 4. Structuri in C/C++

## 4.6 Tablouri ca elemente de structuri

### EXEMPLE

Ex.1. : structura care contine ca elemente o matrice si un sir:

```
struct x
{ int a[10][10];
  float b[10];
} y, *p; p=&y;

...
y.a[3][7]=10;
y.b[1]=2;

...
p->a[3][7]=10;
p->b[1]=2;

...
(*p).a[3][7]=10;
(*p).b[1]=2;
```

# 4. Structuri in C/C++

## 4.7 Structuri imbricate

### DEFINIRE

Declararea structurilor imbricate:

```
struct tip1 { tip var1;  
             tip var2;  
             ...  
             tip vari  
};  
struct tip2 { tip var11;  
             tip var12;  
             struct tip1 var13;  
             ...  
}struct1,struct2, ...structi,...;
```

Accesul la elementele structurii imbricate:

```
structi.var13.vari
```

# 4. Structuri in C/C++

## 4.7 Structuri imbricate

### EXEMPLE

Ex.2. : Structura `adr` de tip `adrese` este imbricata in structura `pers` de tip `angaj`

```
struct adrese { char strada[40];  
                char oras[20];  
                int nr; };  
struct angaj { char nume[30];  
               struct adrese adr  
               float salar;} pers;
```

pentru a atribui valoarea 404 elementului `nr` din structura `adr`:

```
pers.adr.nr=404
```

pentru a citi de la tastatura elementul `strada`:

```
gets(pers.adr.strada);
```



# TEST

1 . Considerand secventa de mai jos :

```
struct data {int luna;  
            int ziua;  
            int an; } birth;
```

Indicati care dintre instructiunile urmatoare are ca efect atribuirea datei de 14.03.1975 variabilei birth:

- a) birth=03.14.1975;
- b) struct birth={"14","03","1975"}
- c) ziua.birth=14; luna.birth=03; an.birth=1975;
- d) birth.luna = 03; birth.ziua = 14; birth.an = 1975;

**Raspuns correct**

**d)**





# TEST

2 . Considerand secventa de instructiuni de mai jos, indicati instructiunea corecta care realizeaza initializarea campului "nr" cu valoarea 1232, prin intermediul pointerului \*p?

```
struct adrese
```

```
{ char strada[40], oras[20]; int nr; };
```

```
struct angaj
```

```
{ char nume[30],prenume[30];
```

```
  struct adrese adr ;
```

```
  float salarbrut;};
```

```
struct angaj pers,*p;p=&pers;
```

**Raspuns correct**

**c)**

- a) p->nr=1232;
- b) p.adr.nr=1232;
- c) p->adr.nr=1232;
- d) p.nr=1232;



# TEST kahoot

Pentru login, introduceti codul afisat pe ecran, in browser la adresa:

<http://kahoot.it>

# 4. Structuri in C/C++

## 4.8 Tablouri de structuri

### DEFINIRE

Definirea unui tablou de structuri

- ❑ se defineste mai intai tipul structurii
- ❑ se declara variabila tablou de structuri

```
struct tip { tip var1;  
            tip var2;  
            ...  
            } tab[max]; // sau tab[max1][max2];
```

### EXEMPLE

Ex.1. : definirea unui tablou de structuri :

```
#include <stdio.h>  
int main()  
{struct adrese {  
    char strada[40];  
    int nr;  
    char oras[20];  
    char judet ; } ;  
struct adrese adr[100];  
adr[2].nr=120;  
printf("nr=%d",adr[2].nr);  
gets(adr[2].strada);  
return 0;}
```

nr=120

# 4. Structuri in C/C++

## 4.8 Tablouri de structuri

### EXEMPLE

Ex.2. : definirea unui tablou de structuri :

```
#include<stdio.h>
struct course
{   int courseno;
    char coursename[25];
};
int main()
{   struct course c[ ] = { {102, "C"},
                          {103, "C++"},
                          {104, "Java"}
                          };
    printf("%d ", c[1].courseno);
    printf("%s\n", c[1].coursename);
    return 0;}
```

103 C++

# 4. Structuri in C/C++

## 4.8 Tablouri de structuri

### EXAMPLE

Ex.3. : definirea unui tablou de structuri :

```
#include<stdio.h>
struct course
{   int courseno;
    char coursename[25];
};
int main()
{   struct course c[ ] = { {102, "C"},
                          {103, "C++"},
                          {104, "Java"}
                          };
    printf("%d ", c[1].courseno); //afiseaza courseno pentru al 2lea element
    printf("%s\n", (*(c+2)).coursename); //afiseaza coursename pentru al 3lea element
    return 0;}

```

103 Java

# 4. Structuri in C/C++

## 4.8 Tablouri de structuri - aplicatii

### EXEMPLE

Ex.4. : Să se scrie un program care definește o variabilă tablou unidimensional de structuri, si realizează urmatoarele operatii:

- citește numărul studenților (n), numele, prenumele și 2 note pentru fiecare student
- calculează media aritmetică a notelor pentru fiecare student
- afișează studenții sortați prin Metoda Bubble Sort în ordinea descrescătoare a mediilor.

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

```
struct student {    char nume[20];  
                   char prenume[20];  
                   double nota1,nota2,media; } s[100], aux[100];
```

```
int i=0,n,k;
```

```
int main ()
```

```
{printf("Introduceti numărul de studenți:");scanf("%d", &n);
```

```
...
```

```
return 0;}
```

continuare

# 4. Structuri in C/C++

## 4.8 Tablouri de structuri - aplicatii

### EXEMPLE

Ex.4. : citirea numărului studenților (dimensiunea n a tabloului), a numelui, prenumelui și a 2 note pentru fiecare student si calculul mediei aritmetice a notelor pentru fiecare student

```
struct student { char nume[20], prenume[20];  
                double nota1,nota2, media; } s[100], aux[100]; //definire tablou structuri
```

```
int i=0,n,k;
```

```
int main (void)
```

```
{printf("Introduceti numărul de studenți:");scanf("%d", &n);
```

```
for (i=0;i<n;i++){ //citire date
```

```
printf("\nIntroduceti numele studentului %d:", i+1); scanf("%s",&s[i].nume);
```

```
printf("Introduceti prenumele studentului %d:", i+1);scanf("%s",&s[i].prenume);
```

```
printf("Introduceti nota1 a studentului %d:",i+1); scanf("%lf", &s[i].nota1);
```

```
printf("Introduceti nota2 a studentului %d:",i+1); scanf("%lf", &s[i].nota2);
```

```
//calcul medie
```

```
s[i].media = (s[i].nota1+s[i].nota2)/2;
```

```
printf("%s %s, nota1=%5.2lf, nota2=%5.2lf, media= %5.2lf \n", s[i].nume,s[i].prenume,  
s[i].nota1,s[i].nota2, s[i].media }; return 0; }
```

continuare

# 4. Structuri in C/C++

## 4.8 Tablouri de structuri - aplicatii

### EXAMPLE

Ex.4. : sortarea studentilor in ordinea mediilor

```
printf("\nStudentii sortati in ordinea descrescatoare a mediilor:\n");
do {k=0;
for (i=0;i<n;i++)
    {if (s[i].media<s[i+1].media) { aux[i]=s[i] ;s[i]=s[i+1];s[i+1]=aux[i];k=1;}}
}
while (k);
for (i=0;i<n;i++)
printf("%s %s, nota1=%5.2lf, nota2=%5.2lf, media= %5.2lf \n", s[i].name, s[i].prenume, s[i].nota1,
s[i].nota2, s[i].media); }
```

```
Studentii sortati in ordinea descrescatoare a mediilor:
Ulaicu Aurel,nota1=10.00, nota2=10.00, media=10.00
Pop Mircea,nota1=10.00, nota2= 9.50, media= 9.75
Albu Ioana,nota1= 8.00, nota2=10.00, media= 9.00
Coldea Mihai,nota1= 7.50, nota2= 8.50, media= 8.00
```



# 4. Structuri in C/C++

## 4.9 Transmiterea structurilor catre functii

### DEFINIRE

Transmiterea elementelor structurii catre functii

- transmitere element cu element
- transmiterea intregii structuri

### EXEMPLE

Ex.1 : transmiterea element cu element catre functii :

```
struct tip
{ char x;
  int y;
  float z;
  char s[10]; } mihai;

...
func(mihai.x);           //preia valoarea de tip caracter din x
func1(mihai.y);         //preia valoarea intrega din y
func2(mihai.z);         //preia valoarea de tip float din z
func4(mihai.s[2]);      //preia valoarea de tip caracter din s[2]
```

# 4. Structuri in C/C++

## 4.9 Transmiterea structurilor catre functii

### EXAMPLE

Ex.2. : transmiterea structurilor catre functii:

```
#include <stdio.h>
struct tip { //definire tip structura global, poate fi accesat de main() si f1()
    int a,b;
    char ch;    };
void f1(struct tip param);
int main(void)
{ struct tip arg;
  arg.a=1000;
  f1(arg); //transmiterea structurii catre functie prin apel de functie
return 0;}
void f1(struct tip param) //definirea functiei
{ printf("%d", param.a);}
```

1000

# 4. Structuri in C/C++

## 4.9 Transmiterea structurilor catre functii

### EXAMPLE

Ex.3. : adunarea a 2 nr. complexe :

```
#include <stdio.h>
struct complex {int re; int im;}; //declaratie globala pentru ambele functii
struct complex suma(struct complex a, struct complex b);
int main(void)
{
    struct complex c1,c2,c3;
    printf("Introduceti primul nr. real re+j*im:\n");
    printf("re="); scanf ("%d", &c1.re);
    printf("im="); scanf ("%d", &c1.im);
    printf("\nIntroduceti al doilea nr. real re+j*im:\n");
    printf("re="); scanf ("%d", &c2.re); printf("im="); scanf ("%d", &c2.im);
    c3=suma(c1,c2);
    printf("\nSuma celor 2 nr. complexe este: %d +j*%d",c3.re,c3.im );return 0;}
struct complex suma(struct complex a, struct complex b)
{
    struct complex c;
    c.re=a.re+b.re; c.im=a.im+b.im; return c;}
```

```
Introduceti primul nr. real re+j*im:
re=2
im=3

Introduceti al doilea nr. real re+j*im:
re=4
im=5

Suma celor 2 nr. complexe este: 6 +j*8
```

# 4. Structuri in C/C++

## 4.9 Transmiterea structurilor catre functii

### EXAMPLE

Ex.4. : adunarea a 2 nr. complexe utilizand **apelul prin referinta** al parametrilor de tip structura:

```
#include <stdio.h>
struct complex {int re; int im;} ;
struct complex suma(struct complex *a, struct complex *b);
int main(void)
{struct complex c1,c2,c3;
printf("Introduceti primul nr. real re+j*im:\n");
printf("re="); scanf ("%d", &c1.re);
printf("im="); scanf ("%d", &c1.im);
printf("\nIntroduceti al doilea nr. real re+j*im:\n");
printf("re="); scanf ("%d", &c2.re); printf("im="); scanf ("%d", &c2.im);
c3=suma(&c1,&c2);
printf("\nSuma celor 2 nr. complexe este: %d +j*%d",c3.re,c3.im ); return 0;}
struct complex suma(struct complex *a, struct complex *b)
{struct complex c;
c.re=a->re+b->re; c.im=a->im+b->im; return c;}
```

```
Introduceti primul nr. real re+j*im:
re=10
im=20
```

```
Introduceti al doilea nr. real re+j*im:
re=30
im=40
```

```
Suma celor 2 nr. complexe este: 40 +j*60
```

# 4. Structuri in C/C++

## 4.10 Alocarea dinamica a memoriei pentru structuri

### EXAMPLE

Ex.1. : alocarea dinamica a memoriei pentru un tablou de structuri:

```
struct pers {char nume[20];  
             char prenume[20];  
             int cod;  
             int varsta; } *dp;
```

...

```
dp=(struct pers *)malloc(n*sizeof(struct pers));  
if (dp==NULL) printf("nu exista suficiente memorie!\n");
```

Alocare dinamica pentru o structura :  $\text{sizeof}(\text{struct pers}) = 20+20+4+4 = 48$  octeti

Alocare dinamica pentru n structuri :  $\text{nxsizeof}(\text{struct pers}) = \text{nx}48$  octeti

# 4. Structuri in C/C++

## 4.10 Alocarea dinamica a memoriei pentru structuri

### EXAMPLE

Ex.1. : alocarea dinamica a memoriei pentru n structuri de tip persoana:

```
#include <stdio.h>
#include <malloc.h>
struct data {int ziua,luna,an; };
struct pers {char nume[15],prenume[20];
             int cod;
             struct data datan;
             int varsta; } *dp;

void citire(struct pers *dp) {
printf("Nume:"); scanf("%s", dp->nume);
printf("Prenume:"); scanf("%s", dp->prenume);
printf("Cod:"); scanf("%d", &dp->cod);
printf("ziua nasterii:"); scanf("%d",
printf("luna nasterii:"); scanf("%d",
printf("anul nasterii:"); scanf("%d",

int main()
{int i,n;
printf("\n nr. angajati:"); scanf("%d",&n);
dp=(struct pers *)malloc(n*sizeof(struct pers));
if (dp==NULL) printf("nu exista suficiente memorie!\n");
printf("Introduceti datele fiecarui angajat:\n");
for (i=0;i<n;i++) {printf("Datele persoanei %d:\n", i+1); citire(dp+i); }return 0;}
```

```
nr. angajati:3
Introduceti datele fiecarui angajat:
Datele persoanei 1:
Nume:Pop
Prenume:Ion
Cod:1121
ziua nasterii:23
luna nasterii:02
anul nasterii:2000
Datele persoanei 2:
Nume:Rusu
Prenume:Ioana
Cod:1120
ziua nasterii:12
luna nasterii:08
anul nasterii:2001
Datele persoanei 3:
Nume:Casian
Prenume:Anca
Cod:1123
ziua nasterii:14
luna nasterii:07
anul nasterii:2002
```



# TEST

1 . Considerand secventa de instructiuni de mai jos, indicati instructiunea corecta care realizeaza initializarea campului cod pentru toate persoanele din tabloul p[], cu valoarea 1232 prin bucla for?

```
struct adrese
```

```
{ char strada[40], oras[20]; int cod; };
```

```
struct angaj
```

```
{ char nume[30],prenume[30];
```

```
  struct adrese adr ;
```

```
  float salarbrut;};
```

```
struct angaj p[100];
```

```
for(i=0;i<100;i++)
```

a) p[i].cod=1232;

b) p[i].adr.cod=1232;

c) p->adr.cod=1232;

d) (p+i).cod=1232;

**Raspuns correct**

**b)**



TEST

Browser telefon :

<http://kahoot.it>

introduceti codul ce va fi afisat pe ecran si numele