

Laborator 6

Definirea structurilor. Structuri și tablouri. Structuri și funcții. Pointeri la structuri

În acest capitol sunt prezentate considerații teoretice și probleme rezolvate privind definirea și utilizarea datelor de tip structură, legătura dintre tablouri și structuri, definirea și utilizarea tablourilor de structuri, legătura dintre structuri și funcții precum și pointeri la structuri.

CONSIDERAȚII TEORETICE

Tipurile de date definite de utilizator se clasifică în limbajul C/C++ în următoarele categorii principale:

- structuri
- uniuni
- enumerari
- typedef
- câmpuri de biți

Definirea structurilor

Structura: este o colecție de variabile ce pot fi de tipuri diferite (spre deosebire de tablouri) reunite sub același nume. La declararea unei structuri se alocă automat memorie pentru fiecare membru al structurii.

Elementele (membri, câmpurile): sunt variabilele din componenta unei structuri care pot fi de orice tip (standard, pointeri, șiruri, alte structuri, etc) iar numele lor pot fi aceleași cu elementele altei structuri din același program.

Declararea tipului structura se realizează după formatul:

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

Declararea variabilei de tip structura se realizează cu instrucțiunea:

```
struct nume_tip_struct lista_variabile_struct;
```

Declararea structurii printr-o singură instrucțiune:

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

unde nume_tip_struct și lista_variabile_struct sunt opționale, dar nu pot lipsi ambele simultan.

Ex.1: Declararea unei structuri de tip student, utilizând declarația de tip separat față de declarația de variabilă .

```
struct student {          // declararea tipului structurii
    int nume[20];
    int age;
    int cod_curs;
    int an_studiu;
    int id_stud; };
struct student stud1,stud2,stud3; //declararea variabilelor
```

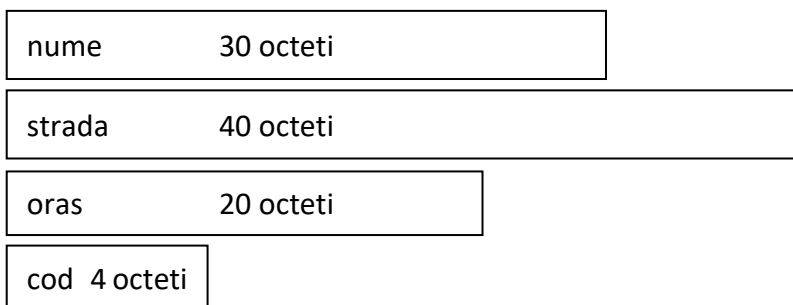
Ex.2: Declararea unei structuri de tip student, într-o singură instrucțiune.

```
struct student {          // declararea tipului structurii
    int nume[20];
    int age;
    int cod_curs;
    int an_studiu;
    int id_stud;
} stud1,stud2,stud3; //declararea variabilelor
```

Ex.3: Declararea unei structuri de tip adresă.

```
struct adrese           // adrese =numele tipului structurii
{ char nume[30]; //elementele sunt declarate intre acolade
  char strada[40];
  char oras[20];
  int cod;
}; // aici se termina declararea tipului structurii
struct adrese adr; //declararea variabilei adr de tipul adrese
```

Pentru această variabilă adr (din Ex.3), de tip structură se vor rezerva $30+40+20+4=94$ octeți de memorie, considerând că tipul char se reprezintă pe câte un octet iar tipul int pe 4 octeți.



Accesul la elementele unei structuri se realizează fie prin **operatorul punct (.)**, astfel:
nume_struct.nume_membru

fie prin intermediul pointerilor la structuri ce vor fi definiți în cele ce urmează.

Inițializarea elementelor unei structuri se realizează similar cu inițializarea șirurilor, utilizând declararea de tip structură, constantele de inițializare fiind separate prin virgulă.

Informația conținută într-o structură poate fi atribuită altei structuri de același tip printr-o singură instrucțiune de atribuire. Astfel, dacă se declară două variabile de același tip structură, a și b ,

instrucțiunea de atribuire este $b=a$; și prin aceasta se înțelege că fiecare din câmpurile structurii a este atribuit câmpului corespunzător din structura b.

Pointeri și structuri

Legătura dintre pointeri și structuri se realizează prin intermediul pointerilor la variabilele de tip structură.

Formatul de **declarare a unui pointer la o structură**:

```
struct tip_struct *adr_pointer
```

unde ***adr_pointer** va fi un pointer către variabila de tip structură de tipul **tip_struct**

Operatorul "săgeată" -> permite accesul la elementele structurii folosind un pointer la această structură.

Ex. : Accesarea elementelor structurii folosind pointeri:

```
struct tip
{ int i
  float bilant;
  } persoana;
struct tip *p; //declara un pointer la structura
p=&persoana; //preia adresa structurii persoana în pointerul p
p->bilant=0; //permite accesul la membrul bilant din structura persoana
p->i++; //incrementeaza membrul i al structurii
```

Pointeri și tablouri.

a)Tablourile ca elemente ale unei structuri

Daca se declară mai multe variabile de tipul structură prezentat mai jos, și un pointer *p la una din aceste variabile, structi:

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

atunci accesul la tabloul tab[] se realizează în două moduri:

```
structi.tab[i]=expresie sau constantă;
p->tab[i]=expresie sau constantă; //cu pointeri
```

iar accesul la tabloul mat[][] se face astfel:

```
structi.mat[i][j]=expresie sau constantă
p->mat[i][j]=expresie sau constantă //cu pointeri
```

b) **Tablourile de structuri** se studiaza în Lab.7.

Structuri imbricate

Formatul de declarare a structurilor imbricate este :

```
struct tip_str1 {
    tip var1;
    tip var2;
    ...
};
struct tip_str2 {
    tip var1;
    tip var2;
    struct tip_str1 structa;
    ...
}struct1,struct2, ...structi,...;
```

Accesul la elementele variabilei **structi** se realizează prin intermediul punctului, în mod analog cu adresarea structurii simple, astfel: **structi.structa.var1;**

Ex.: structuri imbricate (structura adr de tip adrese este imbricată în structura pers de tip angaj)

```
struct adrese { char nume[30];
                char strada[40];
                char oras[20];
                int cod; };
struct angaj { struct adrese adr
               float salar;} pers;
//pentru a atribui valoarea 3400 elementului cod din structura adr:
pers.adr.cod=3400;
```

Structuri și funcții

Transmiterea elementelor structurii către funcții :

- transmitere element cu element către funcții
- transmiterea structurilor întregi către funcții

Ex.: transmiterea element cu element către funcții

```
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 intreaga din y
func2(mihai.z); //preia valoarea de tip float din z
```

```
func3(mihai.s); //preia adresa sirului s
func4(mihai.s[2]); //preia valoarea de tip caracter din s[2]
```

PROBLEME REZOLVATE

Ex.1.: Programul declară o variabilă de tip structură cu format de adresă cu câmpurile nume, strada,nr., ap, local (localitatea) și cod. Se vor citi de la tastatură câmpurile: nume, strada, ap, local și se atribuie prin program valori aleatoare câmpurilor nr și cod. Se cere să se afișeze toate câmpurile acestei structuri.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> int main() { struct adrese{ char nume[30]; char strada[40]; int nr; int ap; char local[20]; int cod; } adr; printf("Numele si prenumele: "); gets(adr.nume); printf("Strada:"); gets(adr.strada); adr.nr=26; printf("apartamentul:"); scanf("%d", &adr.ap); printf("Localitate:"); scanf("%s", &adr.local); adr.cod=400118; 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;}</pre>	<pre>#include <iostream> #include <stdlib.h> #include <string> using namespace std; int main() { string x,y; struct adrese{ char nume[30]; char strada[40]; int nr; int ap; char local[20]; int cod; } adr; cout<<"Numele si prenumele: "; cin.get(adr.nume,30); cin.ignore(); cout<<"Strada: "; cin.get(adr.strada,40); cin.ignore(); adr.nr=26; cout<<"apartamentul: "; cin>>adr.ap; cout<<"Localitate: "; cin>>adr.local; adr.cod=400118; cout<<"\nAti introdus urmatoarele date:"<<endl; cout<<adr.nume<<"\nAdresa:"<<adr.nume<<"," <<adr.strada<<","<<adr.nr<<","ap."<<adr.ap<<en dl; cout<<adr.local<<endl<<"Cod:"<<adr.cod<<endl; return 0;}</pre>

Rezultate:

```
Numele si prenumele: Popa Alessia
Strada:Baritiu
apartamentul:24
Localitate:Cluj-Napoca

Ati introdus urmatoarele date:
Popa Alessia
Adresa:Baritiu,26,ap.24
Cluj-Napoca
Cod:400118
```

Aplicație:

Să se modifice programul astfel încât să se declare 2 variabile de tip adrese: home_adr, office_adr care se inițializează cu date diferite și apoi se afișează.

Ex.2.: Programul este un exemplu de atribuire a unei structuri altei structuri. Se definesc 2 variabile x și y de tip structură cu 2 câmpuri a și b. Se inițializează variabila x cu valori constante și se atribuie structurii y întreaga structură x, afișându-se câmpurile structurii y.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> #include <stdlib.h> int main() {struct { int a; //a,b campurile structurii int b; } x,y; //x,y variabile de tip structura x.a=10; x.b=2; y=x; //atribuie intreaga structura (element cu element) x lui y printf("x.a=%d x.b=%d\n", x.a, x.b); printf("y.a=%d y.b=%d\n", y.a, y.b); return 0;}</pre>	<pre>#include <iostream> #include <stdlib.h> using namespace std; int main() {struct { int a; //a,b campurile structurii int b; } x,y; //x,y variabile de tip structura x.a=10; x.b=2; y=x; //atribuie intreaga structura (element cu element) x lui y cout<<"x.a="<<x.a<<" x.b="<<x.b<<endl; cout<<"y.a="<<y.a<<" y.b="<<y.b<<endl; return 0;}</pre>

Rezultate:

```
x.a=10 x.b=2
y.a=10 y.b=2
```

Aplicație:

Să se modifice programul astfel încât să se afișeze x.a+y.a și y.a+y.b.

Ex.3. Programul ilustrează operațiile de bază (adunare, scădere, înmulțire, împărțire) specifice numerelor complexe, și afișează modulul și argumentele acestora utilizând o variabilă de tip structură cu două câmpuri: partea reală și partea imaginară. Operațiile sunt descrise mai jos.

- **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:**

$$\arg(x + iy) = \begin{cases} \arctg(\frac{y}{x}), \arctg(\frac{y}{x}) \geq 0 \\ \arctg(\frac{y}{x}) + 2\pi, \arctg(\frac{y}{x}) < 0 \end{cases}$$

Varianta in C

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
#define PI 3.1415926
int main()
{struct { 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);
//afisarea primului nr. complex
printf("z=%2.2lf+%2.2lfj\n", z.re, z.im);
//citirea celui de-al 2-lea nr.complex
printf("Introduceti y=c+dj\n");
printf("c= "); scanf("%lf",&y.re);
printf("d= "); scanf("%lf",&y.im);
//afisarea celui de-al 2-lea nr.complex
printf("y=%2.2lf+%2.2lfj\n", y.re, y.im);
//adunarea nr. complexe
printf("adunarea nr. complexe:\n");
printf("(%.2lf+%.2lfj)+(%.2lf+%.2lfj)=%.2f+%.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("(%.2lf+%.2lfj)*(%.2lf+%.2lfj)=%.2f+%.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); //impartirea nr. complexe
printf("impartirea nr. complexe:\n");
if((z.im*z.im+y.im*y.im)>0)
{printf("(%.2lf+%.2lfj)/(%.2f+%.2fj)= \n",z.re,z.im, y.re,y.im);
printf("=%.2lf+%.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 %.2lf + %.2lfj = %.2f\n", z.re, z.im, sqrt(z.re*z.re +z.im*z.im));
printf("modulul lui %.2lf + %.2lfj = %.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(%.2lf+%.2lf j)=%.2lf\n", z.re,z.im, atan2(z.im,z.re));
else printf("arg(%.2lf+%.2lfj)=%.2lf\n", z.re,z.im, atan2(z.im,z.re)+2*PI);
if (atan2(y.im,y.re)>=0.) printf("arg(%.2lf+%.2lf j)=%.2lf\n", y.re,y.im, atan2(y.im,y.re));
else printf("arg(%.2lf+%.2lfj)=%.2lf\n", y.re,y.im, atan2(y.im,y.re) +2*PI);
return 0;}

```

Varianta in C++

```
#include <iostream>
#include<conio.h>
#include <math.h>
#define PI 3.1415926
using namespace std;
int main()
{struct { double re;
          double im;} z,y;
//citirea nr.complexe
cout<<"Introduceti z=a+bj"<<endl;
cout<<"a= "; cin>>z.re;
cout<<"b= "; cin>>z.im;
//afisarea primului nr. complex
cout<<"z="<<z.re<<"+"<<z.im<<"j"<<endl;
//citirea celui de-al 2-lea nr.complex
cout<<"Introduceti y=c+dj"<<endl;
cout<<"c= "; cin>>y.re;
cout<<"d= "; cin>>y.im;
//afisarea celui de-al 2-lea nr.complex
cout<<"y="<<y.re<<"+"<<y.im<<"j"<<endl;
//adunarea nr. complexe
cout<<"adunarea nr. complexe:"<<endl;
cout<<"("<<z.re<<"+"<<z.im<<"j")+("<<y.re<<"+"<<y.im<<"j)"<<endl;
//inmultirea a nr. complexe
cout<<"inmultirea nr. complexe:"<<endl;
cout<<"("<<z.re<<"+"<<z.im<<"j")*("<<y.re<<"+"<<y.im<<"j)"<<endl;
//impartirea nr. complexe
cout<<"impartirea nr. complexe:"<<endl;
if((z.im*z.im+y.im*y.im)>0)
{cout<<"("<<z.re<<"+"<<z.im<<"j")/("<<y.re<<"+"<<y.im<<"j)"<<endl;
cout<<(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)<<"j"<<endl;} else {cout<<"numarator =0"<<endl ;}
//modulul nr. complex
cout<<"modulul lui "<<z.re<<"+"<<z.im<<"j" = "<< sqrt(z.re*z.re +z.im*z.im)<<endl;
cout<<"modulul lui "<<y.re<<"+"<<y.im<<"j" = "<< sqrt(y.re*y.re +y.im*y.im)<<endl;
//argumentul nr.complex
if (atan2(z.im,z.re)>=0.) cout<<"arg("<<z.re<<"+"<<z.im<<"j)"<<atan2(z.im,z.re)<<endl;
else cout<<"arg("<<z.re<<"+"<<z.im<<"j)"<< atan2(z.im,z.re)+2*PI<<endl;
if (atan2(y.im,y.re)>=0.) cout<<"arg("<<y.re<<"+"<<y.im<<"j)"<<atan2(y.im,y.re)<<endl;
else cout<<"arg("<<y.re<<"+"<<y.im<<"j)"<< atan2(y.im,y.re) +2*PI<<endl;
return 0;}
```

Rezultate:**Aplicație:**


```

Introduceti z=a+bj
a= 5.5
b= 4.8
z=5.50+4.80j
Introduceti y=c+dj
c= 1.2
d= 1.1
y=1.20+1.10j
adunarea nr. complexe:
(5.50+4.80j)+(1.20+1.10j)=6.70+5.90j
inmultirea nr. complexe:
5.50+4.80j)*(1.20+1.10j)=1.32+11.81j
impartirea nr. complexe:
5.50+4.80j)/(1.20+1.10j)=1.14+-0.01j
modulul lui 5.50 + 4.80j = 7.30
modulul lui 1.20 + 1.10j = 1.63
arg(5.50+4.80 j)=0.72
arg(1.20+1.10 j)=0.74

```

Să se rezolve aceeași problemă utilizând pointeri la variabilele structură.

Ex.4.: Programul este un exemplu de transmitere a structurilor catre funcții. Se definește funcția de tipărire numită $f1(struct\ tip\ param)$ care are ca și parametru local o variabilă de tip structură.

Varianta in C	Varianta in C++
<pre> #include <stdio.h> #include <stdlib.h> //definirea unui tip de structura global ca sa poata fi "vizibil" //si din functia main si din functia f1 struct tip {int a,b; char ch; }; void f1(struct tip param); //declarare prototip functie utilizator int main() {struct tip arg; arg.a=1000; arg.b=1500; f1(arg); //transmiterea structurii catre functie return 0;} void f1(struct tip param) //definire functie utilizator { printf("arg.a=%d, arg.b=%d\n", param.a, param.b);} </pre>	<pre> #include <iostream> #include <stdlib.h> using namespace std; //definirea unui tip de structura global ca sa poata fi "vizibil" //si din functia main si din functia f1 struct tip {int a,b; char ch; }; void f1(struct tip param); //declarare prototip functie utilizator int main() {struct tip arg; arg.a=1000; arg.b=1500; f1(arg); //transmiterea structurii catre functie return 0;} void f1(struct tip param) //definire functie utilizator { cout<<"arg.a="<<param.a<<"", arg.b="<<param.b<<endl;} </pre>

Rezultate:

```
arg.a=1000, arg.b=1500
```

Aplicație:

Să se modifice programul, astfel încât să se citească un caracter ch de la tastatură și dacă $ch='y'$ să se apeleze o funcție care calculează rezultatul expresiei $\sqrt{a^2 + b^2}$ unde a și b sunt componentele structurii arg .

Ex.5 Programul definește 2 variabile de tip (structură) student cu câmpurile de mai jos și se inițializează cu diverse valori care apoi vor fi afișate utilizând pointeri.

Varianta in C

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main()
{typedef struct {
char name[20];
    int age;
    int year;
    long int id;
    double nota1;
    double nota2;
    double nota3;
    double nota4;} student;
student s1={"Popescu Gheorghe",19,2,2021,10.,9.5,8.4,8.7},
        s2={"Caprar Ioan",18,1,1018,8.5,5.6,5.4,7.},*p1,*p2;
p1=&s1;p2=&s2 ;
printf("Tiparirea studentilor fara pointeri\n");
printf("nume student: %s\nvirsta: %d\nanul de studiu:%d\ncodul identificare: %ld\nmedia: %lf\n", s1.name,
s1.age, s1.year, s1.id,(s1.nota1+s1.nota2+s1.nota3+s1.nota4)/4); //atentie linie continuata!
printf("*****\n");
printf("nume student: %s\nvirsta: %d\nanul de studiu:%d\ncodul identificare: %ld\nmedia: %lf\n", s2.name,
s2.age, s2.year, s2.id,(s2.nota1+s2.nota2+s2.nota3+s2.nota4)/4); //atentie linie continuata!
printf("*****\n");
printf("Tiparirea studentilor cu pointeri\n");
printf("nume student: %s\nvirsta: %d\nanul de studiu: %d\ncodul identificare: %ld\nmedia: %lf\n", p1->name,
p1->age, p1->year, p1->id,(p1->nota1+p1->nota2+p1->nota3+p1->nota4)/4);//atentie linie continuata!
printf("*****\n");
printf("nume student: %s\nvirsta: %d\nanul de studiu:%d\ncodul identificare: %ld\nmedia: %lf\n", p2->name,
p2->age, p2->year, p2->id, (p2->nota1+p2->nota2+p2->nota3+p2->nota4)/4); //atentie linie //continuata!
printf("*****\n");
return 0;}
```

Varianta in C++

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{typedef struct {
char name[20];
    int age;
    int year;
    long int id;
    double nota1;
    double nota2;
    double nota3;
    double nota4;} student;
student s1={"Popescu Gheorghe",19,2,2021,10.,9.5,8.4,8.7},
        s2={"Caprar Ioan",18,1,1018,8.5,5.6,5.4,7.},*p1,*p2;
```

```

p1=&s1;p2=&s2 ;
cout<<"Tiparirea studentilor fara pointeri"<<endl;
cout<<"nume student: "<<s1.name<<endl<<"virsta: "<<s1.age<<endl<<"anul de studiu:
"<<s1.year<<endl<<"codul identificare: "<<s1.id<<endl<<"media:
"<<(s1.nota1+s1.nota2+s1.nota3+s1.nota4)/4<<endl; //atentie linie continuata!
cout<<"*****"<<endl;
cout<<"nume student: "<<s2.name<<endl<<"virsta: "<<s2.age<<endl<<"anul de studiu:
"<<s2.year<<endl<<"codul identificare: "<<s2.id<<endl<<"media:
"<<(s2.nota1+s2.nota2+s2.nota3+s2.nota4)/4<<endl; //atentie linie continuata!
cout<<"*****"<<endl;
cout<<"Tiparirea studentilor cu pointeri"<<endl;
cout<<"nume student: "<<p1->name<<endl<<"virsta: "<<p1->age<<endl<<"anul de studiu: "<<p1-
>year<<endl<<"codul identificare: "<<p1->id<<endl<<"media: "<<(p1->nota1+p1->nota2+p1->nota3+p1-
>nota4)/4<<endl; //atentie linie continuata!
cout<<"*****"<<endl;
cout<<"nume student: "<<p2->name<<endl<<"virsta: "<<p2->age<<endl<<"anul de studiu: "<<p2-
>year<<endl<<"codul identificare: "<<p2->id<<endl<<"media: "<<(p2->nota1+p2->nota2+p2->nota3+p2-
>nota4)/4<<endl; //atentie linie //continuata!
cout<<"*****"<<endl;
return 0;}

```

Rezultate:

```

Tiparirea studentilor fara pointeri
nume student: Popescu Gheorghe
virsta: 19
anul de studiu:2
codul identificare: 2021
media: 9.150000
*****
nume student: Caprar Ioan
virsta: 18
anul de studiu:1
codul identificare: 1018
media: 6.625000
*****
Tiparirea studentilor cu pointeri
nume student: Popescu Gheorghe
virsta: 19
anul de studiu: 2
codul identificare: 2021
media: 9.150000
*****
nume student: Caprar Ioan
virsta: 18
anul de studiu:1
codul identificare: 1018
media: 6.625000
*****

```

Aplicație:

Să se modifice programul astfel încât să se inițializeze datele prin citire de la tastatură și să se afișeze doar studenții care au media peste 8.50.

Ex.6. Programul preia de la tastatură o dată calendaristică și afișează data din ziua următoare. Data calendaristică este declarată sub forma unei structuri cu trei câmpuri.

Varianta in C	Varianta in C++
<pre>#include <stdio.h> struct data { //variabile globale int zi, luna, an;}; int zi[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31,30,31 }; struct data azi, maine; void getdataazi(void); //prototip functie void getdataazi(void) { int valid = 0; while(valid == 0) {printf("Introduceti anul (2000-2019)-->"); scanf("%d", &azi.an); if((azi.an < 1990) (azi.an > 2019)) printf("Invalid an\n"); else valid = 1;} valid = 0; while(valid == 0) { printf("Introduceti luna (1-12)-->"); scanf("%d", &azi.luna); if((azi.luna < 1) (azi.luna > 12)) printf("Invalid luna\n"); else valid = 1;} valid = 0; while(valid == 0) {printf("Introduceti ziua (1-%d)-->", zi[azi.luna-1]); scanf("%d", &azi.zi); if((azi.zi < 1) (azi.zi > zi[azi.luna-1])) printf("Invalid zi\n"); else valid = 1; } } int main() {getdataazi(); maine = azi; maine.zi++; if(maine.zi > zi[maine.luna-1]) { maine.zi = 1; maine.luna++; if(maine.luna > 12) { maine.an++; maine.luna = 1; } } printf("Data de maine:%02d:%02d:%02d\n",maine.zi, maine.luna, maine.an); return 0;}</pre>	<pre>#include <iostream> using namespace std; struct data { //variabile globale int zi, luna, an;}; int zi[] = { 31, 28, 31, 30, 31, 30, 31, 31,30,31,30,31 }; struct data azi, maine; void getdataazi(void); //prototip functie void getdataazi(void) { int valid = 0; while(valid == 0) {cout<<"Introduceti anul (2000-2019)-->"; cin>>azi.an; if((azi.an < 1990) (azi.an > 2019)) cout<<"Invalid an"<<endl; else valid = 1;} valid = 0; while(valid == 0) { cout<<"Introduceti luna (1-12)-->"; cin>>azi.luna; if((azi.luna < 1) (azi.luna > 12)) cout<<"Invalid luna"<<endl; else valid = 1;} valid = 0; while(valid == 0) {cout<<"Introduceti ziua (1-<<zi[azi.luna-1]<<")-->"; cin>>azi.zi; if((azi.zi < 1) (azi.zi > zi[azi.luna-1])) cout<<"Invalid zi"<<endl; else valid = 1; } } int main() {getdataazi(); maine = azi; maine.zi++; if(maine.zi > zi[maine.luna-1]) { maine.zi = 1; maine.luna++; if(maine.luna > 12) { maine.an++; maine.luna = 1; } } cout<<"Data de maine:"<<maine.zi<<":"<<maine.luna<<":"<< maine.an<<endl; return 0;}</pre>

Rezultate:

```
Introduceti anul (2000-2019)-->2019
Introduceti luna (1-12)-->01
Introduceti ziua (1-31)-->05
Data de miine:06:01:2019
```

Aplicație:

Să se modifice programul astfel încât să se testeze dacă anul este bisect

Ex.7. Programul definește o structură de tip angajat al unei companii (angaj) care conține o altă structură imbricată de tip adresă (adrese). Programul citește de la tastatură datele angajaților și calculează impozitul pe baza salariului brut (considerat cunoscut), după formula: $\text{impozit} = 40\%$ din salariu brut.

Varianta in C
<pre> #include <stdio.h> int main() {struct adrese { char strada[40]; char oras[20]; int cod; }; struct angaj {char nume[30]; struct adrese adr ; float salarbrut; float impozit; float salarnet ; }; struct angaj pers1; printf("Introduceti nume angajat:"); scanf("%30s", pers1.nume); printf("\nIntroduceti adresa angajat:"); printf("\nIntroduceti strada: "); scanf("%s", pers1.adr.strada); printf("Introduceti orasul: "); scanf("%s", pers1.adr.oras); printf("Introduceti cod: "); scanf("%d", &pers1.adr.cod); printf("\nSalar brut pentru angajat: "); scanf("%f", &pers1.salarbrut); printf("\n"); pers1.impozit=pers1.salarbrut*0.4; pers1.salarnet=pers1.salarbrut-pers1.impozit; printf("Nume:%s\nAdresa:\n%s,%s\nCod numeric:%4d\n", pers1.nume, pers1.adr.strada,pers1.adr.oras, pers1.adr.cod); printf("Salar brut: %.2f\nImpozit: %.2f\nRest de plata: %.2f\n", pers1.salarbrut, pers1.impozit,pers1.salarnet); return 0;} </pre>
Varianta in C++
<pre> #include <iostream> using namespace std; int main() {struct adrese { char strada[40]; char oras[20]; int cod; }; struct angaj {char nume[30]; struct adrese adr ; float salarbrut; float impozit; float salarnet ; }; struct angaj pers1; cout<<"Introduceti nume angajat:"; cin>>pers1.nume; cin.ignore(); cout<<"\nIntroduceti adresa angajat:"<<endl; cout<<"Introduceti strada: "; cin.get(pers1.adr.strada,40); cin.ignore(); cout<<"Introduceti orasul: "; </pre>

```

cin.get(pers1.adr.oras,20);
cin.ignore(); cout<<"Introduceti cod: ";
cin>>pers1.adr.cod; cout<<"\nSalar brut pentru angajat: ";
cin>>pers1.salarbrut;
cout<<endl;
pers1.impozit=pers1.salarbrut*0.4;
pers1.salarnet=pers1.salarbrut-pers1.impozit;
cout<<"Nume:"<<endl<<pers1.numere<<"Adresa:"<<endl<<pers1.adr.strada<<","<<pers1.adr.oras<<endl<<"Cod
numeric:"<< pers1.adr.cod<<endl;
cout<<"Salar brut: "<<pers1.salarbrut<<endl<<"Impozit:"<<pers1.impozit<<endl<<"Rest de
plata:"<<pers1.salarnet<<endl; return 0;}

```

Rezultate:

```

Introduceti nume angajat:Popescu
Introduceti adresa angajat:
Introduceti strada: G.Baritiu
Introduceti orasul: Cluj-Napoca
Introduceti cod: 400118

Salar brut pentru angajat: 4000

Nume:Popescu
Adresa:
G.Baritiu,Cluj-Napoca
Cod numeric:400118
Salar brut: 4000.00
Impozit:1600.00
Rest de plata:2400.00

```

Aplicație:

Să se modifice programul astfel încât să se calculeze și afișeze salarul net dacă impozitul este calculat cu formula $SalarBrut * 30\% - 500$ pentru $SalarBrut < SalarBrutMinimEconomie$ și $SalarBrut * 30\% + 500$ dacă $SalarBrut \geq SalarBrutMinimEconomie$ (2080 lei/luna).

PROBLEME PROPUSE

1. Se consideră un magazin de echipamente electronice, în care fiecare echipament reprezintă un articol specificat prin următoarele câmpuri:

- Denumire (alfanumeric, max.30 caractere). Ex. : Video Recorder Panasonic SV500
- Cod (alfanumeric, max 6 caractere). Ex. A254G9
- Garanție (numeric întreg, max. 2 cifre) . Ex. 3
- Preț (numeric real , max. 10 cifre). Ex. 4 500 000

Să se scrie programul C/C++ în care să se citească 3 articole de la tastatură iar tipărirea acestora să se realizeze, utilizând numai câmpul denumire și prețul calculat în EURO (1 EURO= aprox.40.000 lei).

2. Se consideră un magazin de tehnică de calcul, în care fiecare produs reprezintă un articol specificat prin următoarele câmpuri:

- Denumire (alfanumeric, max.30 caractere). Ex. : PC Compaq P910
- Tip (alfabetic, max.10 caractere). Ex. calculatoare
- Caracteristici (alfabetic, max.30 caractere). Ex. 800MHz, 10 GB HDD,CDROM
- Garanție (numeric întreg, max. 2 cifre) . Ex. 3
- Preț (numeric real , max. 5 cifre). Ex. 950

Prețul e specificat în EURO și se introduce de la tastatură. Să se scrie programul C/C++ în care să se introducă de la tastatură 3 articole de tipul specificat mai sus, să se calculeze prețul echivalent în lei și să se afișeze articolele, utilizând formatul de mai jos:

<i>Denumire</i>	<i>Pret (LEI)</i>
<i>PC Compaq P910</i>	<i>28 500 000 lei</i>
<i>HP LaserJet 5000</i>	<i>15 000 000 lei</i>

...

3. *Se consideră o bibliotecă, în care fiecare carte sau revistă reprezintă un articol specificat prin următoarele câmpuri:*

- *Titlu (alfanumeric, max.30 caractere). Ex. : The Theory of Electric Circuits*
- *Autori (alfabetic, max 30 caractere). Ex. Michael Douglas*
- *Editura (alfanumeric, max.30 caractere). Ex. Teora*
- *ISBN (alfanumeric, max 14 caractere). Ex. 973-9204-98-8*
- *Pret (numeric real , max. 10 cifre). Ex. 350 000*

Să se scrie programul C/C++ care realizează citirea a 2 articole de la tastatură iar tipărirea acestora să se realizeze utilizând numai câmpul denumire, prețul în lei și în EURO (1 EURO= aprox.40.000 lei).

4. *Se consideră o papetărie, în care fiecare produs reprezintă un articol specificat prin următoarele câmpuri:*

- *Denumire (alfanumeric, max.15 caractere). Ex. : Caiet A4*
- *Caracteristici (alfanumeric, max.20 caractere). Ex. File veline*
- *Pret (numeric real , max. 10 cifre). Ex. 55 000*

Sa se scrie programul C/C++ care realizează citirea a 5 articole de tipul specificat mai sus , de la tastatură iar tipărirea articolelor să se realizeze utilizând toate câmpurile citite iar la sfârșit să se afișeze prețul total al articolelor.