

STRUCTURI IMBRICATE. VECTORI DE STRUCTURI



Membrii unei structuri de date neomogene pot fi, ei înșiși de tip structură.

```
struct Data{
    int zi, luna, an;
};
struct Persoana{
    char nume[21], prenume[21], sex;
    int varsta;
    Data data_angajarii;
    double salariu;
};
Persoana A;
```

Accesul la datele membru ale variabilei A, de tip Data se face utilizând de două ori operatorul de selecție.

Ex: A.data_angajarii.an

Considerând declarațiile următoare, care dintre următoarele referiri este corectă din punct de vedere sintactic ?

```
1. struct punct{float x,y};  
2. struct cerc  
3. {float raza;  
4. punct centru;} c;
```



Varianta 1 `c.punct.y`

Varianta 2 `c.raza.punct`

Varianta 3 `c.centru.x`

Varianta 4 `c.y.centru`

VECTORI DE STRUCTURI

Se poate declara un tablou cu elemente de tip structură.

Exemplu:

```
struct elev  
{char nume [20], prenume [20];  
float media;} e[30];
```

Citirea a "n" componente ale vectorului "e" se poate face astfel:

```
cout<<"n="; cin>>n;  
for (i=0; i<n; i++)  
{cout<<"Nume:" ;  
cin>>e[i].nume;  
cout<<"Prenume:" ;  
cin>>e[i].prenume;  
cout<<"Media:" ;  
cin>>e[i].media;}
```

FISA DE LUCRU

Variabila **e**, declarată alăturat, memorează informații despre un eveniment din anul 2019 (numărul de ordine și data desfășurării sale), iar variabila **d** memorează o dată calendaristică din același an. Scrieți o expresie care are valoarea 1 dacă și numai dacă data memorată în variabila **d** este anterioară datei desfășurării evenimentului corespunzător variabilei **e**. (6p.)

```
struct tdata
{   int zi,luna;
}d;
struct eveniment
{   int nr;
    struct tdata dev;
}e;
```

Expresiile de mai jos au ca valori numere naturale, reprezentând următoarele informații memorate pentru un eveniment din anul 2019: data desfășurării sale (ziua și luna) și un identificator. Scrieți definiția unei structuri cu eticheta **eveniment**, care să permită memorarea informațiilor menționate pentru un eveniment, și declarați corespunzător variabila **e**, de acest tip.

e.data.zi e.data.luna e.id (6p.)

Pentru un număr complex se memorează următoarele date: partea reală și partea imaginară (numere reale). Variabila **z** memorează simultan date pentru fiecare dintre cele 20 de numere complexe. Știind că expresia C/C++ de mai jos are valoarea sumei dintre partea reală și partea imaginară ale primului număr complex dintre cele precizate, scrieți definiția unei structuri cu eticheta **complex** care să permită memorarea datelor unui număr complex, și declarați corespunzător variabila **z**.

z[0].pre+z[0].pim (6p.)

(4p.)

```
}m[20];
```

- b. `m[0].substanta.cod`

Știind că expresiile C/C++ de mai jos au ca valori luna (un număr natural din intervalul $[1,12]$) și anul în care au fost măsurate temperaturi (număr natural), respectiv prima temperatură medie înregistrată (un număr real), scrieți definiția unei structuri cu eticheta **meteo**, care permite memorarea informațiilor precizate, și declarați corespunzător variabila **m**.

m.an

(6p.)

FISA DE LUCRU

Variabila **m** memorează simultan, pentru fiecare dintre cele 20 de mașini oferite spre închiriere, următoarele date: anul fabricației mașinii (număr natural) și tipul de carburant al acesteia (șir de maximum 50 de caractere).

Știind că expresiile C/C++ de mai jos au ca valori anul fabricației celei de a treia mașini, respectiv tipul de carburant al acesteia, scrieți definiția unei structuri cu eticheta **masina**, care permite memorarea datelor despre o mașină, și declarați corespunzător variabila **m**.

m[2].an

m[2].carburant

(6p.)

1. În declararea alăturată, în câmpurile **s** și **d** ale structurii sunt memorate extremitatea stângă, respectiv dreaptă ale fiecăruia dintre cele 20 de intervale închise. O expresie C/C++ a cărei valoare este extremitatea stângă a primului interval este:

(4p.)

```
struct interval  
{ float s,d;  
}v[20];
```

a. **v[0].s**

b. **s.v[0]**

c. **v(0).s**

d. **s(0).v**