

Tipul înregistrare- aplicații

SET 1 – Definiri de structuri sau variabile de tip înregistrare

1. Să se definească o structură cu numele **mașina**, care va conține **marca** mașinii (un șir de maxim 20 de caractere), **capacitatea cilindrică** (un număr întreg) și **data înmatriculării** (formată din zi, lună an) astfel încât pentru o variabilă **m** definită de tip mașină să se poată accesa câmpurile astfel:

m.marca	m.capacitate	m.data.zi	m.data.an
---------	--------------	-----------	-----------

2. Să se definească o structură cu numele **medicament**, pentru care trebuie memorate următoarele informații: **denumirea** (un șir de maxim 20 de caractere), **prețul** (un număr real) și **data expirării** (formată din zi, lună an) astfel încât pentru o variabilă **m** definită de tip medicament să se poată accesa câmpurile astfel:

m.denumire	m.pret	m.datae.zi	m.datae.an
------------	--------	------------	------------

3. Să se definească o structură cu numele **segment**, care va conține două puncte notate **A** și **B** care formează capetele segmentului. Pentru fiecare punct în parte se vor memora **coordonatele carteziene (x, y)** (numere reale). Variabila **m** va fi definită de tip segment astfel încât să se poată accesa câmpurile structurii astfel:

m.A.x	m.A.y	m.B.x	m.B.y
-------	-------	-------	-------

4. Să se definească o structură **cerc**, care memorează **raza** (număr întreg) și **coordonatele centrului** cercului **O(x, y)** (numere reale). Definiți două variabile **C1** și **C2** care memorează caracteristicile a două cercuri astfel încât să se poată accesa câmpurile structurii astfel:

C1.O.x	C1.O.y	C1.raza	C2.O.y
--------	--------	---------	--------

5. Definiți o structură cu numele **complex**, care va memora **partea reală** și **partea imaginară** a unui număr complex (ambele numere întregi). Definiți apoi o variabilă cu numele **a** care va memora informațiile celor 10 numere complexe dintr-o aplicație astfel încât să se poată accesa câmpurile structurii astfel:

A[0].pr	A[0].pim
---------	----------

6. Să se definească o structură cu numele **meteo**, care să memoreze **temperatura maximă** și **minimă** în fiecare zi dintr-o lună din an. Variabila **Mai** păstrează temperaturile înregistrate în luna mai din 2020. Cum trebuie declarate datele astfel încât accesul să fie:

Mai.t[1].min	Mai.t[1].max	Mai.t[2].min	Mai.t[2].min
--------------	--------------	--------------	--------------

7. Să se definească o structură cu numele **abonat**, care memorează **numele și prenumele** fiecărui client (șir de maxim 30 de caractere) **suma de plată** de pe luna curentă (număr real) și două date calendaristice: data limită până la care poate fi achitată factura curentă și data achitării facturii anterioare. Pentru fiecare dată calendaristică se va memora luna și anul. Firma are maxim 100 de clienți. Datele de facturare ale clienților vor fi memorate într-un vector, astfel încât accesul să fie:

a[1].nume	a[2].datae.an	a[3].dataf.luna	a[4].suma
-----------	---------------	-----------------	-----------

8. Să se definească o structură cu numele **fracție** care memorează x - numitorul și y - numărătorul unei fracții aritmetice. Trebuie memorate maxim 100 de fracții. Definiți datele astfel încât accesul la componentele variabilei f să fie:

f.a[0].x	f.a[0].y	f.a[1].x	f.a[1].y
----------	----------	----------	----------

SET 2 – Expresii

1. Se consideră următoarea declare :

```
struct medicament {
    char denumire[30];
    char afectiune[100]; ///tipul afectiunii medicale
    float preț;
}x;
```

Scrieți o expresie care este adevărată dacă și numai dacă:

- Denumirea medicamentului are prima literă "P"
 - Dacă este recomandat pentru "raceala"
 - Dacă prețul medicamentului aparține intervalului [a, b], unde a, b sunt două numere întregi $a < b$.
2. Se consideră următoarea declare :

```
struct Elev {
    char nume[20],doptionala[10];
    float notaR, notaM, notaO;
} x[30];
```

Scrieți o expresie care este adevărată dacă și numai dacă:

- Verifică dacă nota la limba română este mai mare decât nota la matematică pentru elevul din poziția a doua(vectorul este indexat de la 1)
 - Dacă primii doi elevi au notele la matematică egale
 - Dacă primii doi elevi au ales aceeași disciplină opțională
3. Se consideră următoarea declare :

```
struct Produs {char denumire[20], categorie[20]; ///litere mici
    float cantitate, preț;
}x, y;
```

Scrieți o expresie care este adevărată dacă și numai dacă:

- Cele două produse încep cu aceeași literă
 - Dacă prețul produsului x este mai mare decât prețul produsului y
 - Dacă produsul x mai este în stoc(cantitate diferită de zero)
4. Se consideră următoarea declare :

```
struct punct{
    float x;
    float y;
}A;
```

Scrieți o expresie care este adevărată dacă și numai dacă:

- Punctul A se află pe prima bisectoare;

- b. Punctul A se află pe abscisă;
 - c. Punctul A este în originea sistemului de coordonate xOy.
5. Se consideră următoarea declarație :

```
struct punct{
    float x;
    float y;
}A,B;
```

Scrieți o expresie care este adevărată dacă și numai dacă:

- a. Segmentul AB este paralel cu axa Oy;
 - b. Are unul din capete în origine;
 - c. Se află pe axa Ox.
6. Se consideră următoarea declarație:

```
struct punct{
    float x;
    float y;
};
struct cerc{
    punct O; ///coordonatele centrului cercului
    float raza;
}C1,C2;
```

Scrieți o expresie care verifică dacă cercurile C1 și C2 sunt:

- a. exterioare
 - b. tangente exterior/interior
 - c. concentrice
7. Se consideră următoarea declarație:

```
struct punct{
    float x;
    float y;
}P;
struct cerc{
    punct O; ///coordonatele centrului cercului
    float raza;
}C1;
```

Scrieți o expresie care este adevărată dacă și numai dacă

- a. Centrul cercului C1 coincide cu punctul P
 - b. Punctul P este în interiorul cercului C1
 - c. Punctul P este în exteriorul cercului C1
8. Se consideră următoarea declarație:

```
struct punct{
    float x;
    float y;
}P;

struct segment{
    punct A,B;
}S;

struct cerc{
    punct O; ///coordonatele centrului cercului
    float raza;
}C;
```

Scrieți o expresie care este adevărată dacă și numai dacă:

- Raza cercului **C** este egală cu lungimea segmentului **S**
- Punctul **P** este mijlocul segmentului **S**
- Punctul **P** coincide cu centrul centrului cercului **C**

SET 2 – secvențe de instrucțiuni

- Se consideră următoarea declarație:

```
struct punct{  
    float x;  
    float y;  
};  
struct cerc{ punct O; ///coordonatele centrului cercului  
    float raza;  
}C1,C2;
```

Dacă avem citite coordonatele centrelor și raza pentru două cercuri **C1** și **C2** scrieți o secvență de program care verifică dacă cele două cercuri au două puncte comune(sunt secante).

- Se consideră un număr complex $z=re + i*im$. Fie următoarea declarație a unui nou tip de date:

```
struct complex{  
    int pre;    /// partea reală a unui număr complex  
    int pim;    /// partea imaginară a unui număr complex  
}z1, z2, z;
```

Dacă se cunoaște partea reală și partea imaginară a două numere complexe scrise în forma algebrică **z1** și **z2** scrieți o secvență de program care atribuie variabilei **z** suma celor două numere complexe date.

- Se consideră un număr complex $z=re + i*im$. Fie următoarea declarație a unui nou tip de date:

```
struct complex{  
    int pre;    /// partea reală a unui număr complex  
    int pim;    /// partea imaginară a unui număr complex  
}z1, z2, z;
```

Dacă se cunoaște partea reală și partea imaginară a două numere complexe scrise în forma algebrică **z1** și **z2** scrieți o secvență de program care atribuie variabilei **z** produsul celor două numere complexe date.

- Fie următoarea declarație de tip:

```
struct fractie{  
    int x;    /// numaratorul  
    int y;    /// numitorul unei fractii de forma x/y  
}f1, f2, f;
```

Scrieți o secvență de program care calculează și atribuie variabilei **f** suma dintre cele două fracții **f1** și **f2**. Nu este nevoie de citirea datelor.

- Se consideră următoarea declarație:

```
struct masina {  
    char marca[20], culoare[10];  
    int anF;  
    float preț;} x;
```

Scrieți o secvență de program care mărește cu 20 % prețul mașinii **x** și afișează pe ecran marca, culoarea și noul preț.

6. Se consideră următoarea declare:

```
struct produs {char denumire[20], categorie[20];  
                float cantitate, preț;  
                }x[101];  
  
int n;
```

Scrieți o secvență de program care calculează valoarea totală a celor n produse din depozit.(vectorul este indexat de la 1).

7. Se consideră următoarea declare:

```
struct punct{  
    float x;  
    float y;  
}P;  
  
struct segment{  
    punct A,B;  
}S;  
  
struct cerc{  
    punct O; ///coordonatele centrului cercului  
    float raza;  
}C;
```

Scrieți o secvență de program care calculează diametrul cercului C, lungimea segmentului S, si distanța punctului P față de originea sistemului de coordonate xOy.

8. Se consideră următoarea declare:

```
struct punct{  
    float x;  
    float y;  
}P;  
  
struct dreptunghi{  
    punct A,B;  
}D;
```

unde:

- în variabila P sunt memorate coordonatele unui punct în planul xOy
- în variabila D se memorează coordonatele vârfului stânga sus și dreapta jos a unui dreptunghi care are laturile paralele cu axele

Scrieți o secvență de program care calculează aria dreptunghiului și verifică dacă punctul P este unul din vârfurile dreptunghiului dat.