

## LABORATOR 4

### 1. Tipuri de date stucturate

- colectie de valori de tipuri diferite
- declaratie:

```
struct [id_tip] {  
    tip_camp_1 id_camp_1;  
    tip_camp_2 id_camp_2;  
    ...  
    tip_camp_n id_camp_n;  
} [lista_identificatori_de_tip_struct];
```

- daca *id\_tip* lipseste, structura se numeste anonima;
- daca *lista\_identificatori\_de\_tip\_struct* lipseste dar are *id\_tip*, se pot defini ulterior variabile de tipul structurii prin:  
**struct** <*id\_tip*> <*lista\_identificatori\_de\_tip\_struct*>

- Exemplu:

```
struct student {  
    char cnp[13];  
    char nume[15], prenume[15];  
    int varsta;  
    float medie_an[5];  
} student1, student2;  
  
struct student student3;
```

- pentru a referi un membru al unei variabile de tip struct de foloseste „.”
- cuvantul cheie **enum** se foloseste pentru a defini enumeratii – ansamblu de constante intregi asociate cate unui identificator; implicit, sirul valorilor e crescator cu pasul 1, iar prima valoare e 0;

```
int ore_lucru[7];  
enum zile_sapt {D, L, Ma, Mc, J, V, S} zi;  
for (zi = L; zi <=V; zi++) ore_lucru[zi] = 8;
```

- uniunile se declara ca si structurile, dar se inlocuieste **struct** cu **union**; ele memoreaza in acelasi spatiu de memorie mai multe variabile suprapuse;
- cu ajutorul cuvintului cheie typedef se asociaza un nume/sinonim cu un tip de date;

```
typedef unsigned int Nat;
typedef struct student student;
```

## 2. Directive catre preprocesare

- directiva #define se foloseste pentru:
  - a atribui un nume unei constante speciale
 

```
#define PI 3.1415926
```

```
#define TRUE 1
```

```
#define FALSE 0
```
  - a defini macroinstructiuni:
 

```
#define print_float(n) printf("%.2f",n);
```

```
#define par(n) ((n%2)? 1:0)
```

## 3. Tablouri unidimensionale

- colectie de valori de acelasi tip
- declaratie:

**tip *identificator*[dimensiune];**

- Exemplu:

```
int a[10];
char s[100];
```

- initializare:

```
int a[4] = {1, 2, 3}; //a[3] va fi 0
char c[] = {'a', 'b', 'c', '\0'};
char c[] = "abcd";
char c[3] = {'a', 'b', 'c'};
char c[3] = "abc";
```

- **Obs.: Intotdeauna componentele unui vector vor fi alocate de la pozitia 0 la n-1, inclusiv. Astfel, de exemplu, parcurgerea folosind instructiunea for va fi:**

```
for (i=0; i<n; i++) {...}
```

## **PROBLEME:**

1. a). Sa se construiasca o structura ce contine urmatoarele date despre candidatii la admitere: nr\_legitimatie, nume, nota\_mate, nota\_info, nota\_bac, medie, admis (Y/N), buget (Y/N).  
  
b). Sa se defineasca o macroinstructiune ce calculeaza media de admitere dupa regula: 80% media la examen, 20% media de bac.  
  
c). Sa se defineasca o constanta pentru pragul minim de promovabilitate egala cu 5.  
  
d). Sa se scrie o functie care citeste datele unui candidat, in afara de medie, admis si buget, si le adauga unui vector al tuturor candidatilor, pastrand ordinea alfabetica. Media si promovabilitatea vor fi calculate folosind definitiile de la punctele b) si c). Numarul de candidati este citit de la tastatura.
  - se va folosi functia **strcmp(s1,s2)** ce returneaza un numar:
    - negativ, daca s1 este mai mic decat s2 dpdv al continutului
    - zero, daca s1 este identic cu s2
    - pozitiv, daca s1 este mai mare decat s2 dpdv al continutului  
e). Sa se scrie o functie care completeaza campul „buget” cu Y sau N dupa regula: primii 75% (rotunjit in jos) dintre candidatii admisi, in ordinea mediilor, sunt la buget (Y), restul la taxa (N) sau nu au promovat examenul de admitere (lasati campul gol).  
  
f). Sa se scrie o functie care afiseaza datele candidatilor in functie de optiunea aleasa: toti candidatii (alfabetic), cei admisi la buget, cei admisi la taxa, cei respinsi (ordonati descrescator dupa medie). (meniu cu switch)
2. Creati o structura pentru a reprezenta un numar complex. Realizati operatiile de adunare, inmultire, impartire, ridicare la putere (citita de la tastatura). Afisati rezultatele sub forma  $a + bi$  cu mesaje corespunzatoare si folositi functii pentru fiecare operatie.

### **Observatii:**

- Nu se vor folosi alte functii de lucru cu siruri de caractere in afara de strcmp;
- Nu se vor folosi pointeri
- Toate afisarile trebuie sa contina mesaje corespunzatoare