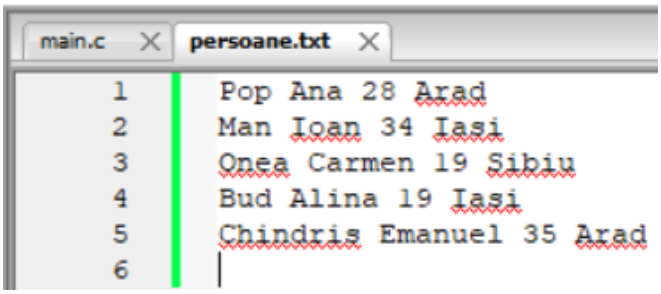
Laborator 10. Rezolvări și explicații

Să se rezolve prin realizarea programului C următoarea problemă:

1. Se primește la intrare un fișier text care pe fiecare linie conține următoarele informații: *nume persoană, prenume persoană, vârsta (întreg), localitate.* Să se citească întreg fișierul de intrare în program. Să se afișeze numărul de persoane existente din fiecare vârstă citită. Să se afișeze numărul de persoane din fiecare localitate.

Rezolvare:

Să presupunem că fișierul text numit **persoane.txt** primit la intrare deține pe fiecare linie, separate printr-un spațiu, următoarele informații despre persoane (în această ordine: nume prenume vârstă localitate):



În urma citirii fișierului de mai sus, va fi nevoie să stocăm informații despre 5 persoane. De exemplu, pe prima linie în fișier avem informațiile despre o persoană cu numele complet Pop Ana, vârsta de 28 de ani și localitatea Arad.

Observație: Fișierul **persoane.txt** poate fi stocat pe partiția C: sau D: a mașinii dvs., poate fi un fișier accesat la distanță, sau un fișier aflat pe un suport extern, însă, pentru simplificare, puteți stoca fișierul în proiectul Code::Blocks curent, folosind opțiunea File→New→Empty file.

În cele ce urmează vom explica pas cu pas logica programului ca soluție a problemei descrie în enunț, axându-ne pe porțiunile de cod aferente.

Pentru ca programul nostru C să fie capabil de a stoca informații structurate despre persoane, și anume: numele, prenumele, vârsta și localitatea de domiciliu a acestora, mai întâi vom defini un nou tip de dată complexă, indusă de cuvântul cheie `struct`. Codul de mai jos declară structura *persoana*, alcătuită din patru câmpuri: *nume*, *prenume*, *varsta* și *localitate*, precum și variabila `sir[100]` de acest tip.

```
struct persoana {
     char nume[20];
     char prenume[20];
     int  varsta;
     char localitate[25];
} sir[100];
```
(1)

Definiția structurii *persoana* se încheie cu declararea variabilei `sir[100]`, ce reprezintă un șir de 100 de persoane. În acest șir vom stoca informațiile citite din fișierul de intrare.

Structura definită la punctul **(1)** este echivalentă cu:

```
struct persoana {
      char nume[20];
      char prenume[20];
      int  varsta;
      char localitate[25];
};
struct persoana sir[100];
```

(2)

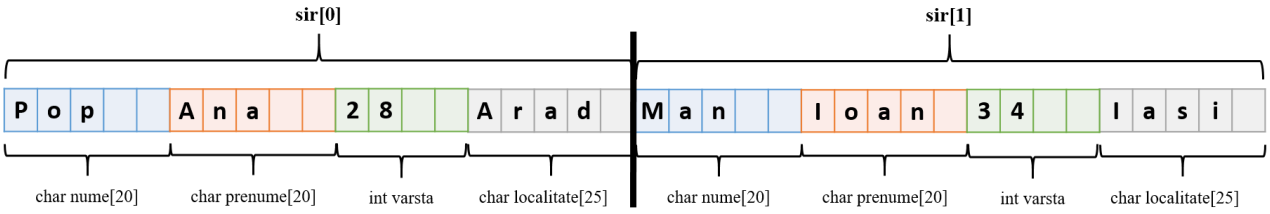
Codul de la punctul **(2)** separă definirea tipului/structurii persoană de declararea unei variabile șir de acest tip. În programul nostru putem opta pentru oricare dintre aceste stiluri de scriere, întrucât ambele sunt corecte.

Apoi, pentru a citi și stoca informațiile despre persoane din fișierul de intrare, definim o funcție numită *citeste_inregistrari()*, care conține următoarele instrucțiuni:

```
int citeste_inregistrari(FILE *fp)
{
    int i=0;
    while(fscanf(fp,"%s %s %d %s", &sir[i].nume,
                                   &sir[i].prenume,
                                   &sir[i].varsta,
                                   &sir[i].localitate)==4)

        i++;
    return i;
}
```

În instrucțiunile de mai sus folosim funcția *fscanf* pentru a citi cu format, linie cu linie, informațiile despre persoane din fișier. Indexul *i* numără înregistrările/liniile din fișier. În urma citirii unei noi înregistrări din fișier, variabila *i* se poziționează în șirul de persoane *sir[]* pentru a stoca informațiile citite în variabilele *nume, prenume, varsta* și *localitate*. În memorie, șirul de persoane este un șir de structuri de tip *persoana.*



După finalizarea sarcinii de citire a informațiilor din fișier, sirul *sir[]* va conține informațiile corespunzătoare tuturor persoanelor. În figura de mai sus, este reprezentat în memorie sirul *sir[]*, evidențiind pentru primele 2 poziții dir șir, valorile variabilelor *nume, prenume, varsta* și *localitate.*

Pentru a afișa numărul de persoane existente din fiecare vârstă citită, va trebui să sortăm șirul după *vârstă*. În fișierul **persoane.txt** de mai sus, există 2 persoane cu vârsta de 19 ani, respectiv câte 1 persoană cu vârsta de 28, 34 și 35 de ani. Vom crea o functie numită *sorteaza_varsta()*, care foloseste ca și argument variabila *n* =dimensiunea șirului de persoane. În această funcție, folosim o variabila booleana *schimbare* (inițial 0) prin care presupunem că inițial șirul este sortat crescător după vârstă.

Parcugem șirul de persoane, comparând vârsta a două persoane alăturate *sir[i]* cu *sir[i+1]*. Dacă vârsta persoanei de pe poziția *i* din șir este mai mare decât vârsta persoanei de pe poziția *i+1* din șir atunci interschimbăm aceste persoane, folosind o variabilă auxiliară *tmp* care stochează temporar o copie a informațiilor persoanei de pe pozitia *i* din șir. Executăm toate interschimbările necesare, rearanjând în șir toate persoanele, astfel încât la final să obținem șirul sortat crescător după valoarea variabilei *varsta*.

Observație: A se remarca faptul că referim câmpul *varsta* din structură folosind operatorul ".". Câmpul *varsta* selectat se comportă ca o variabilă întreagă obișnuită putând să îi aplicăm toate operațiile care se pot aplica variabilelor de tipul întreg.

```
void sortare_varsta(int n)
{
   struct persoana tmp;
   int i,schimbare;
   do
   {
        for(schimbare=0,i=0;i<n-1;i++)
        {
             if(sir[i].varsta > sir[i+1].varsta)
             {
                  tmp=sir[i];
                  sir[i]=sir[i+1];
                  sir[i+1]=tmp;
                  schimbare=1;
             }
        }
   } while(schimbare==1);
}
```

Pentru a afișa numărul de persoane din fiecare localitate, vom proceda similar ca mai sus, creând o funcție auxiliară numită *sortează_localitate()* pentru a ordona lexicografic persoanele în șir în funcție de localitatea de domiciliu. Funcția *sortează_localitate()* va fi asemănătoare cu funcția *sorteaza_varsta()*, însă cu diferența că vom folosi funcția predefinită *strcmp()* ca să comparăm șirurile de caractere aferente localităților a două persoane din *sir[i]*, respectiv *sir[i+1]*.

Programul C complet este următorul:

```c
#include <stdio.h>
#include <stdlib.h>
struct persoana { //definim structura cu numele persoana
	char nume[20];
	char prenume[20];
	int varsta;
	char localitate[25];
} sir[100]; //variabila sir[100] va fi de tip structură persoana
//funcţia pentru citirea înregistrărilor din fişier
int citeste_inregistrari(FILE *fp)
{
	int i=0;
	while(fscanf(fp,"%s %s %d %s", &sir[i].nume, &sir[i].prenume,
				&sir[i].varsta, &sir[i].localitate)==4)
		i++;
	return i;
}
//funcţia de ordonare a persoanelor după vârstă
void sortare_varsta(int n)
{
     struct persoana tmp;
     int i,schimbare;
     do
     {
         for(schimbare=0,i=0;i<n-1;i++)
         {
              if(sir[i].varsta > sir[i+1].varsta)
              {
                   tmp=sir[i];
                   sir[i]=sir[i+1];
                   sir[i+1]=tmp;
                   schimbare=1;
              }
         }
     }while(schimbare==1);
}
// funcţia de ordonare a persoanelor după localitate
void sortare_localitate(int n)
{
     struct persoana tmp;
     int i,schimbare;
     do
     {
         for(schimbare=0,i=0;i<n-1;i++)
         {
              if(strcmp(sir[i].localitate,sir[i+1].localitate)==1)
              {
                   tmp=sir[i];
                   sir[i]=sir[i+1];
                   sir[i+1]=tmp;
                   schimbare=1;
              }
         }
     } while(schimbare==1);
}
```

```
void afiseaza_inregistrari(int n)
{
     int i;
     for(i=0;i<n;i++)
          printf("%s %s %d %s\n",sir[i].nume,sir[i].prenume,sir[i].varsta,
                 sir[i].localitate);
}
int main()
{
     int nrinregistrari,numar=1,i;
     FILE *fp;
     fp=fopen("persoane.txt","r"); //deschidem fișierul în mod citire
     if(fp==NULL) // în caz de eroare
     {
          printf("Nu am reusit deschiderea fisierului!\n");
          return -1;
     }
     nrinregistrari=citeste_inregistrari(fp);
     sortare_varsta(nrinregistrari); //sortăm persoanele din sir după vârstă
     printf("\nInregistrarile sortate dupa varsta:\n");
     afiseaza_inregistrari(nrinregistrari);
     //numărăm persoanele din fiecare vârstă citită
     i=0;
     do
     {
          while((i+1<nrinregistrari)&&(sir[i].varsta==sir[i+1].varsta))
              i++,numar++;
          printf("\nAferent varstei %d ani avem %d inregistrari\n",
                  sir[i].varsta, numar);
          i++;
          numar=1;
     }while(i<nrinregistrari);

     sortare_localitate(nrinregistrari); //sortăm persoanele după localitate
     printf("\nInregistrarile sortate dupa localitate:\n");
     afiseaza_inregistrari(nrinregistrari);
     // numărăm persoanele din fiecare localitate
     i=0;
     do
     {
          while((i+1<nrinregistrari)&&(strcmp(sir[i].localitate,
                                          sir[i+1].localitate)==0))
               i++,numar++;
          printf("\nAferent localitatii %s avem %d inregistrari\n",
                  sir[i].localitate, numar);
          i++;
          numar=1;
      } while (i<nrinregistrari);

      fclose(fp); //închidem fișierul
      return 0;
}
```

Output:

```
Numarul de inregistrari este 5

Inregistrarile sortate dupa varsta:
Onea Carmen 19 Sibiu
Bud Alina 19 Iasi
Pop Ana 28 Arad
Man Ioan 34 Iasi
Chindris Emanuel 35 Arad

Aferent varstei 19 ani avem 2 inregistrari

Aferent varstei 28 ani avem 1 inregistrari

Aferent varstei 34 ani avem 1 inregistrari

Aferent varstei 35 ani avem 1 inregistrari

Inregistrarile sortate dupa localitate:
Pop Ana 28 Arad
Chindris Emanuel 35 Arad
Bud Alina 19 Iasi
Man Ioan 34 Iasi
Onea Carmen 19 Sibiu

Aferent localitatii Arad avem 2 inregistrari

Aferent localitatii Iasi avem 2 inregistrari

Aferent localitatii Sibiu avem 1 inregistrari
```