

Matemàtica computacional i analítica de dades

Algorítmia i combinatòria en grafs...

Curs 2024–25

2 Treball amb taules

En aquesta sessió de treball veurem alguns exemples relatius al tractament de dades guardades primer en una taula i, a la Pràctica 5, en forma de llista enllaçada.

El codi següent llegeix una *llista de notes* en la que les dades de cada línia són un identificador numèric (NIU) i quatre valors decimals entre 0 i 10 en una taula. Tot i que desconexem la longitud de la taula, no treballa amb memòria dinàmica i fa una reserva inicial de posicions. En cas de que la taula que es llegeix sigui més llarga que la memòria reservada, dona un missatge d'error.

```
1 # include <stdio.h>
2 # include <stdlib.h>
3
4 #define MAX_DADES 400
5
6 typedef struct Dada{
7     int niu;
8     float notes[5];
9 } Alu;
10
11 float mitjana(float [],int );
12
13 int main(){
14     FILE *dades;
15     int n, i, j, lrg = 0;
16
17     Alu alumnes[MAX_DADES];
18
19     dades=fopen("Llista.txt","r");
20     if(dades==NULL){
21         printf("\nNo s'ha accedit al fitxer de dades\n");
22         return 1;
23     }
24
25     while(!(fscanf(dades,"%i",&n)==EOF)){
26         if(lrg==MAX_DADES){
27             printf("No hi ha prou espai reservat per guardar totes les
dades\n");
28             return 2;
29         }
30         alumnes[lrg].niu=n;
31         for(i=0;i<4;i++){
32             fscanf(dades,"%f",&alumnes[lrg].notes[i]);
33             fgetc(dades); //llegeix i descarta els ; i el \n
34         }
35         alumnes[lrg].notes[4]=mitjana(alumnes[lrg].notes,4);
36         lrg++;
37     }
38     fclose(dades);
39     for(j=0;j<lrg;j++){
40         printf("%d | ",alumnes[j].niu);
41         for(i=0;i<4;i++){
42             printf("%5.1f",alumnes[j].notes[i]);
43         }
44         printf(" |%6.2f",alumnes[j].notes[4]);
```

```

45     printf("\n");
46 }
47 printf("\nS'ha llegit informacio de %d linies.\n\n",lrg);
48
49 return 0;
50 }
51
52 float mitjana(float dades[],int n){
53     int i;
54     float m=0.;
55     for(i=0;i<n;i++)
56     {
57         m+=dades[i];
58     }
59     return m/n;
60 }

```

2.1 Exercicis

Afegiu a les primeres línies del programa i en format comentari els vostres Nom, Cognom i NIU.

El nom dels programes que contenen el codi ha de ser de la forma PrXExY.c, on X fa referència a la pràctica i Y a l'exercici. Per exemple, el nom del programa de l'apartat següent hauria de ser Pr2Ex211.c.

Exercici 2.1.1: Modifiqueu el codi anterior per a que, enlloc de reservar 400 posicions de memòria independentment del contingut del fitxer, miri primer quantes línies té el fitxer i faci la reserva necessària per a poder guardar les dades.

Exercici 2.1.2: Feu que, després de llegir el fitxer (i per tant, després de fer la reserva de memòria), el programa afegeixi a la taula alumnes un alumne nou amb dades:

1234567 5.6 6.2 4.3 8.6

Observació: Heu de tenir en compte que s'ha de modificar la reserva de memòria, i ho podeu fer amb la funció `realloc`.

Exercici 2.1.3: Creeu una funció `imprimirtaula` que, donada la taula `alumnes` i el nombre de casos, mostri per pantalla la mateixa sortida que mostra el programa inicial.

Exercici 2.1.4: Creeu una funció `ordenataula` que, a partir de la taula `alumnes` i un segon argument amb una taula `alumnesord` buida, guardi a la segona taula la mateixa informació que hi ha a la primera, però amb les entrades ordenades per NIU (la funció necessitarà el nombre d'alumnes com a paràmetre). Feu que el programa mostri per pantalla aquesta segona taula.

Instruccions finals

Quan acabeu la pràctica, feu el lliurament dels fitxers de codi (que tenen els noms de la forma PrXExY.c segons el que hem indicat anteriorment) a través del Campus Virtual des de l'apartat de lliuraments de l'assignatura.