
Llenguatges de Programació

Curs 2006 – 07 / Bloc #2 - Pràctica #2


Notes importants:

- Per fer els exercicis d'aquesta pràctica necessiteu haver llegit a partir de l'apartat 5 del tema 6 i fins al final d'aquest tema del manual de C. Fixeu-vos bé en tots els exemples que hi ha explicats en el manual.
- Recordeu que si no us dóna temps d'acabar tots els exercicis durant la sessió de pràctiques és molt important que els acabeu després pel vostre compte.

Exercici#1> Heu de saber que quan declareu un array (`TipusArray NomArray[Long]`), `NomArray` es pot considerar com un punter que us està donant l'adreça del primer element de l'array. Així al primer element de l'array s'hi pot accedir fent `*NomArray`, al segon `*(NomArray+1)` i així successivament fins a l'últim que s'hi accedeix fent `*(NomArray+Long-1)`. La notació que heu estat fent servir fins ara, la indexada, és una manera equivalent (i més intuïtiva, no ho negarem) de fer això mateix. Per convèncer-vos d'això:


Declareu un vector del tipus que vulgueu i de la longitud que vulgueu. Demaneu a l'usuari que l'empleni i, usant les dues notacions, mostreu el contingut per pantalla. Us ha de donar el mateix!!

Exercici#2> Escriviu una funció que agafi com a paràmetre d'entrada una cadena de caràcters i calculi el percentatge de lletra *a* que hi ha a la cadena. Per fer-ho la funció haurà de calcular primer el nombre de lletres que conté la cadena (sense comptar els espais!) i el nombre de lletres *a* (o *A*) que conté la cadena, i després haurà de calcular el corresponent percentatge. Escriviu també el programa principal que llegeixi la cadena, cridi a la funció i mostri el resultat per pantalla. Utilitzeu l'operador d'autoincrement (`p++`) per accedir als elements de la cadena dins de la funció.

 **NOTA:** Recordeu que per a recórrer la funció, no heu d'utilitzar l'estructura iterativa `for`, sinó la `do/while`, ja que, en principi, no sabeu quina longitud té la cadena que us ha entrat l'usuari...

Exercici #3> Feu un programa en què l'usuari pugui entrar vectors de dimensió variable i calcular-ne la seva mitja. En aquest exercici hi 3 tasques ben diferenciades: crear el vector amb la longitud que ha introduït l'usuari, emplenar-lo i calcular-ne la mitja. Implementeu-les en 3 funcions que haurien de tenir un prototipus similar a aquest:

- `void CreaVector(float**,int*);` `float* CreaVector(int*);`
- `void EmplenVector(float*,int);`
- `float Mitja(float*,int);`

 **NOTA: NO** oblideu d'alliberar la memòria que heu reservat abans de finalitzar el programa.

Exercici #4> Definiu un nou tipus de variable que es digui `n_complex` que permeti guardar nombres complexos (és a dir, haurà de tenir 2 camps, un per la part real i l'altre per la part imaginària). Utilitzant aquesta estructura (que ha d'estar definida de forma global), escriviu procediments (tipus `void`) per fer les següents operacions: suma de dos nombres complexos, la

resta de dos nombres complexos, la multiplicació de dos nombres complexos i el càlcul del mòdul d'un nombre complex. Com que els procediments no retornen res hauran de rebre per referència la variable (o les variables) on es guardarà el resultat.

Feu també el codi dels següents procediments i funcions i utilitzeu-los en el vostre programa principal:

char menu (void): ha de fer que surti per pantalla el menú i ha de permetre a l'usuari que triï una opció i retornar-la.

*mostra_nombre (*n_complex)*: rep un apuntador a un *n_complex* i mostri per pantalla el nombre.

**n_complex llegir_nombre (void)*: crea un apuntador a un *n_nombre*, demana per pantalla un nombre, el posa en el contingut del *n_nombre* i retorna l'apuntador a l'estructura.



Exercici que heu d'entregar abans de la sessió vinent a través del Campus Virtual

Considerem el següent tipus de dada:

```
typedef struct{
    char NomGrup[20];
    char NomDisc[50];
    char Discografica[20];
    int any;
}CD;
```

I el següent punter (local del main) *CD *Discoteca*. Feu un programa que consti de:

- Una funció, *CreaDiscoteca* que ha de desenvolupar la mateixa tasca que la funció *CreaVector* de l'exercici 3.
- Una funció *EmplenaDiscoteca*, que demani les dades per pantalla.
- Una funció, *CercaGrupINFO*, a la qual li passeu el vector apuntat per *Discoteca*, la longitud que hagi entrat l'usuari i un nom de grup llegit de pantalla. Aleshores aquesta funció recorre tot el vector en busca d'un (o més d'un) camp que contingui el valor especificat. Si es troba, mostrem la resta de dades per pantalla. En cas contrari es mostra un missatge.

Aquest procés es repetirà tantes vegades com vulgui l'usuari.



NOTA: NO oblideu d'alliberar la memòria que heu reservat!!!



**Empty your memory,
with a free()...
like a pointer!**

**If you cast a pointer to a integer,
it becomes the integer,
if you cast a pointer to a struct,
it becomes the struct...
The pointer can overflow...,
or can crash...
Be a pointer my friend...**