

Taller de Programación II

Lenguaje de Programación C





Repaso de algunos conceptos básicos.

Variables

Variables Locales: Son las declaradas en una función o dentro de una estructura de control

- La variable pierde el valor en sucesivas llamadas
- Una vez declaradas hay que inicializarlas

Variables Globales: El ámbito se extiende desde su declaración hasta el final del programa

- Se declaran al principio del programa. Deben inicializarse y cualquier función puede acceder a ellas.
- Si la declaración no es anterior a la función, cuando se la usa debe acompañarse con la expresión `extern` antes del nombre.

Variables estáticas:

- solo se antepone la palabra `static` antes de la declaración, pueden ser locales o globales, solo cambia en que el valor perdurará durante toda la ejecución del programa.



Repaso de algunos conceptos básicos

Referencias y operadores de memoria

¿Qué es un puntero?

Una variable que guarda la dirección de memoria de otra variable.

¿Por qué punteros y referencias a variables?

Porque C utiliza valores como input de funciones, o sea, no envía la variable en sí sino su contenido. Esto hace casi imposible que se modifiquen los contenidos de las variables, para lograrlo, usamos las referencias a las memoria o los punteros a esta para que el valor guardado allí se modifique.



Repaso de algunos conceptos básicos

Referencias y operadores de memoria

Los operadores que se utilizan son dos:

`*` y `&`

El uso va a depender del contexto:

`int * pEntero` → declaramos un puntero a entero. Deben inicializarse ***NULL***.

`int x; &x` → aquí con el ampersand obtenemos la dirección de memoria donde está la variable entera `x`

`*pEntero = 10;` → con el asterisco delante de un puntero ya declarado accedo al espacio de memoria para guardar el valor del entero.



Repaso de algunos conceptos básicos

Punteros y arrays

Los arrays están emparentados con los punteros. Cuando se los declara, se reserva una cantidad de memoria. El nombre del array en sí mismo es la dirección de memoria del primer elemento.

Por lo tanto, es lícito hacer lo siguiente:

```
int x[10], *pArreglo;
```

```
pArreglo = x;
```



Repaso de algunos conceptos básicos

Aritmética de punteros

Como hablamos de memoria y tamaños de variables o sea, porciones de memoria, con el puntero podemos movernos dentro de un arreglo

`pArreglo + 0` -> Apunta a la posición inicial del vector

`pArreglo + 1` -> Apunta a la segunda posición del vector

`pArreglo + i` -> Apunta a la posición $i+1$ del vector

`pArreglo = &v[9];` -> p apunta ahora a la última posición (décima) del vector

`p - 1` -> Apunta a la novena posición del vector

`p - i` -> Se refiere a la posición $9 - i$ en v



Repaso de algunos conceptos básicos Importante para no confundir

Dijimos que entre el array y el puntero hay una relación. La vimos con ejemplos simples.

¡¡Es importante no pensar que son sinónimos!!

```
int v[10], *a;  
  
v = a;    /*Está mal!!!*/  
  
v++;     /*Está mal!!!*/
```