Laboratorio 3

15 de abril de 2019

Punteros
Tipo: Evaluado

1. Introducción

En este laboratorio se evaluará el manejo de punteros.
Motivación: *Pointers in C are easy and fun to learn. Some C programming tasks are performed more easily with pointers, and other tasks, such as dynamic memory allocation, cannot be performed without using pointers. So it becomes necessary to learn pointers to become a perfect C programmer. Let's start learning them in simple and easy steps.*[1]

2. Problemas

2.1. Quick Sort

Para este laboratorio, usted debe crear una función que compare dos vectores, y luego usar la función qsort de stdlib.h para ordenar un arreglo de vectores
A continuación hay un ejemplo para un arreglo de enteros. Puede modificar este programa para que ordene vectores de un arreglo o hacer uno desde cero, como más le acomode.

[1] **Pointers in C.** `https://www.tutorialspoint.com/cprogramming/c_pointers.htm`

```c
#include <stdio.h>
#include <stdlib.h>

int values[] = { 88, 56, 100, 2, 25 };

int cmpfunc (int * a, int * b) {
   return ( *a - *b );
}
// Esta función retorna un número negativo si a < b, cero si son iguales y un número positivo si a > b.

int main () {
   int n;

   printf("Before sorting the list is: \n");
      for( n = 0 ; n < 5; n++ ) {
         printf("%d ", values[n]);
      }

      qsort(values, 5, sizeof(int), cmpfunc);
   // values: Array {88, 56, 100, 2, 25}
   // 5: Cantidad de elementos de 'values' (len(values))
   // sizeof(int): Tamaño de los valores de 'values'
   //     Si fuera un arreglo de chars, sería sizeof(char)
   // cmpfunc: Es la función definida en linea 6. A qsort le interesa el signo del número que retorna (negativo, cero, positivo).

   printf("\nAfter sorting the list is: \n");
   for( n = 0 ; n < 5; n++ ) {
      printf("%d ", values[n]);
   }

   return(0);
}
```

Cada vector tendrá tres dimensiones, en caso de empate en la primera dimensión, deberá verificar la segunda, si en esta también hay empate, use la tercera dimensión, en caso de tres empates, da lo mismo que vector vaya primero.
Tenga presente que a la función que ordene la lista tiene que recibir los valores por referencia.
Imprima en consola la lista de valores antes y después de ordenarlas.
Ejemplo:

Antes	Después
(2,3,4)	(2,3,1)
(9,4,5)	(2,3,4)
(2,3,1)	(9,4,5)

3. Detalles

En el archivo "copiar_esto.txt"puede encontrar el arreglo que debe ordenar, tiene la cantidad de vectores (array_size) y el tamaño de cada vector (dimentions).
Archivos que debe entregar: lab3_Apellido_Nombre.c