ALGORITMOS Y ESTRUCTURAS DE DATOS



AEDD - Guía Práctica 14: Punteros

Ejercicio 1: Para cada uno de los siguientes enunciados, escriba el código correspondiente. Suponer que se han declarado las siguientes variables:

```
float num1 = 3.1415926, num2 = 0.12345;
```

- (a) Declarar un puntero a float llamado pnum.
- (b) Asignar la dirección de la variable num1 a pnum.
- (c) Imprimir el valor de la variable apuntada por pnum.
- (d) Asignar el valor de la variable apuntada por pnum a la variable num2.
- (e) Imprimir el valor de num2.
- (f) Imprimir la dirección de num2 y la dirección almacenada en pnum. ¿Es la misma dirección?

Ejercicio 2: Para cada uno de los siguientes enunciados, escriba el código correspondiente.

- (a) Declarar un arreglo de tipo float, llamado *numeros* e inicialice los elementos con los siguientes valores: {0.1, 0.2, 0.3, 0.4, 0.5}. Suponga que la constante SIZE ha sido definida como 5.
- (b) Declarar un puntero pnum que apunte a una variable de tipo float.
- (c) Imprimir los elementos del arreglo utilizando subíndices de arreglos.
- (d) Idem (c) pero utilizando únicamente la variable pnum.
- (e) Suponiendo que pnum apunta al principio del arreglo, ¿Cuál es la dirección referenciada por (pnum+3)? ¿Y su valor?
- (f) Suponiendo que pnum apunta al cuarto elemento del arreglo (numeros[3]). ¿A qué elemento del arreglo apunta (pnum-2)?

Ejercicio 3: Crear un array de 10 números aleatorios y después un array de 10 punteros donde cada uno apunte a los números anteriores.

Ejercicio 4: Dado un vector de 10 elementos = {1, 2, 3, 4, 4, 7, 8, 9, 5, 4}, escribir un programa en C (haciendo uso de punteros) que muestre las direcciones de memoria de cada elemento del vector.

Ejercicio 5: Dada una cadena, invertirla haciendo uso de la aritmética de punteros.

Ejercicio 6: Una aplicación de gestión de fotografías en tu teléfono celular tiene definido el catálogo de fotos de la siguiente forma:

ALGORITMOS Y ESTRUCTURAS DE DATOS



```
#define SIZE_NAME

struct picture_info
{
    char name[SIZE_NAME];
    int date time;
} pictures[SIZE];
```

¿Qué tamaño tiene esta estructura de datos? La aplicación necesita crear una segunda tabla del mismo número de elementos, pero en lugar de tener los datos de las fotos quiere tener los punteros a los datos de las fotos. En otras palabras, es una tabla con idéntico número de elementos que la anterior, pero sus elementos no son estructuras sino punteros a las correspondientes estructuras de la tabla pictures. Escribe la declaración y el código para rellenar la tabla.

http://www.it.uc3m.es/pbasanta/asng/course_notes/ch05s11.html

Ejercicio 7: Una imagen puede representarse como una matriz de puntos (pixels) donde para cada uno de ellos se debe indicar su tono de gris. Dicho tono es un número entre 0 y 255, donde 0 representa negro, 255 blanco y los valores intermedios dan el tono de gris correspondiente.

Como las imágenes ocupan un gran espacio de memoria, se pensó en usar una estructura como la siguiente:

	pu	nto 1								pto. 64
Fila	-									
Fila	_	e e								
Fila	1	97			02	3	1		3 3	
Fila		0	j		Ü					
Fila		97			9			9		
480					Sec			Sec.		

en donde Fila1, Fila2, FilaN son punteros a los pixels de cada línea.

Estamos trabajando con un monitor que tiene una resolución de 480x640 pixels (o puntos). Se solicita que usted:

- 1- Defina la estructura de datos correspondiente.
- 2- Calcule el tamaño de la estructura al momento de comenzar el programa.
- 3- Escriba un módulo que le permita ingresar las primeras 100 filas de la pantalla (no olvide que cada fila está formada por 640 puntos). Indique el tamaño de la estructura una vez que el módulo se haya ejecutado.
- 4- Cuál es el tamaño de la estructura completa, es decir, cuando se hayan almacenado los 480 x 640 puntos.