

Práctico 6

Asignación dinámica de memoria

La teoría general para este práctico puede consultarse en Tutorial de C.

Nota: Recuerde controlar en cada caso si hay memoria suficiente para hacer la asignación y devolver la memoria solicitada antes de final del programa.

1. Realice un programa que muestre por pantalla el valor almacenado, la dirección y el tamaño de los punteros definidos para los siguientes datos:

- a. un entero
- b. un char
- c. un float
- d. una cantidad n de enteros con n

2. Solicite memoria para una cantidad n de enteros la cual es ingresada por el usuario. Además, inicialice cada uno de los enteros con los números 1 al n respectivamente.

3.- Dada la siguiente estructura:

```
typedef struct { char a;  
                int b; } Info;
```

Hacer un programa en donde:

- a. Pida memoria para una estructura de tipo Info.
- b. Asignar valores a cada campo de la estructura.
- c. Mostrar por pantalla la información guardada.

4. Dada la siguiente definición:

```
typedef struct p { char nomb[20];  
                  char ape[20];  
                  int edad; } persona;
```

- a. Solicite memoria para n estructuras. La cantidad de estructuras a almacenar es menor igual a 10 y es ingresada por el usuario.
- b. Realice una función que permita cargar una estructura y la almacene en la memoria asignada.
- c. Realice una función que muestre por pantalla los datos almacenados en cada una de las estructuras.
- d. Realice una función que permita al usuario cargar n estructuras utilizando la función definida previamente. Realice los controles necesarios

5. Dada la estructura definida en el práctico 4 ejercicio 4 (salud):

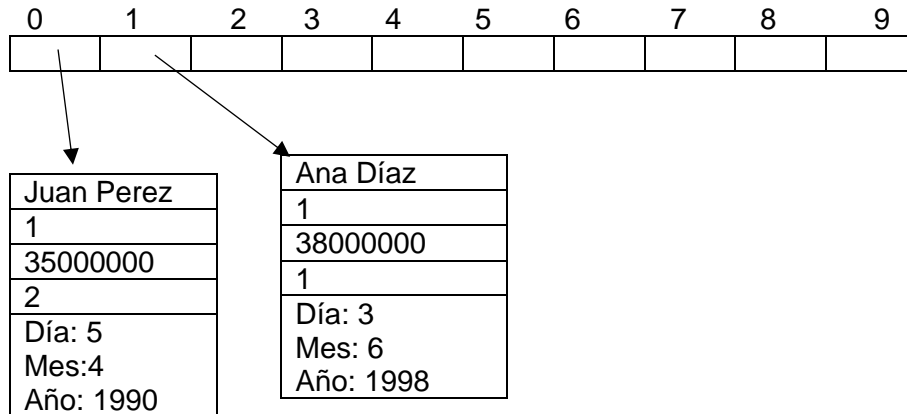
- a. Pida memoria para almacenar dicha estructura.
- b. Almacene los datos en la estructura.
- c. Muestre los datos cargados por pantalla
- d. Solicite memoria para almacenar 50.000 estructuras e. Permita cargar n estructuras

6. Dadas las siguientes estructuras:

```
typedef struct { int dia, mes, año ; } Fecha;  
typedef struct { char nom[30];  
                int TipoDoc;           (1=DNI, 2=PAS, 3=CF, 4=CIFE)  
                int NroDoc[8];  
                int Hijos ;  
                Fecha Nacim; } Empleado;
```

Y la siguiente variable Empleado *arr[10]; (arr es un arreglo de punteros a Empleado).

arr:



Se pide hacer una función que:

- Permita ingresar información de a lo sumo 10 empleados.
- Muestre la información guardada.
- Liste el Nombre de los empleados mayores de 40 años.
- Retorne la cantidad de empleados sin hijos. Integar todas las funciones en un Menú de opciones y liberara memoria al final.

Ejercicios complementarios

1. Indique si las siguientes sentencias son correctas o incorrectas. Justifique en cada caso:

- char *a;
- a= "hola"
- b[]="estas?";
- a=(char *)malloc(sizeof(b));
- a=(char *)malloc(strlen(b)+1);
- a= "chau"
- (*a)= "chau"
- strcpy(*a,b);
- strcpy(a,b);
- free(b)
- free(a)

2. Dadas las siguientes declaraciones:

```
struct X { int len;  
          char *str; } *p, ar[5];
```

Asumiendo p=ar describir las siguientes expresiones:

- ++(*p.len)
- *(++p).len
- *(p++).len
- (*p.str)
- (*p.str)++
- (*p.str)++
- ((*p++).str)

3. Redefina la estructura del ejercicio 5 de manera tal que el campo nombre y apellido de la estructura sean char *.

- ¿Cómo sería la sentencia de asignación de memoria?
- ¿Qué modificaciones debe realizar en las funciones realizadas en el ejercicio anterior?
- ¿Cuál es el tamaño necesario para almacenar la estructura?