

PRÁCTICA III**PUNTEROS*****Ejercicio 3.1***

Programar una función que reciba como parámetros dos punteros a int y que intercambie los valores de las variables apuntadas por dichos punteros.

Ejercicio 3.2

Programar una función que reciba como parámetro un arreglo de int y una posición de dicho arreglo. Luego la función deberá retornar la dirección de memoria de la variable que se encuentra en esa posición del arreglo. ¡Atención! Esta función se resuelve en una sola línea y no requiere más de 15 caracteres (6 de los cuales son “return”).

Ejercicio 3.3

Programar una función que redondee al entero más próximo el valor contenido en una variable externa de tipo double.

Ejercicio 3.4

Dada la estructura:

```
typedef struct
{
    int numero, valor;
    char palo;
}
t_carta;
```

programar una función que modifique el contenido de 3 variables externas de este tipo, de manera que las cartas almacenadas en éstas queden ordenadas según su valor en el juego del “Truco”.

Ejercicio 3.5

Dada la estructura:

```
typedef struct
{
    int a, b, c, d;
}
t_cuatro;
```

programar una función que modifique el contenido de una variable externa de este tipo, de manera que los valores contenidos en cada campo queden ordenados. Es decir, el campo “a” debe tener el valor más bajo, el campo “b” el segundo más bajo y así sucesivamente.

Ejercicio 3.6

Dada la estructura:

```
typedef struct
{
    int n, o, p, q, r, s, t, u, v, w, x, y, z;
    char p1[100], p2[200], p3[300], p4[400], p5[500];
    double a, b, c, d, e, f, g, h, i, j, k, l, m;
}
t_varios;
```

programar una función que intercambie los valores de 2 variables de este tipo, las cuales son externas a la función. ¡Atención! Esta función requiere menos de 5 (cinco) instrucciones.

MEMORIA DINAMICA**Ejercicio 3.7**

Programar una función que permita ingresar una frase por teclado hasta presionar ENTER y que la almacene en la memoria sin desperdiciar espacio. Este nuevo espacio de memoria (conteniendo el texto ingresado) debe ser remitido al exterior de la función para su posterior utilización.

Ejercicio 3.8

Programar una función que copie un string cualquiera a un espacio de memoria dinámico sin desperdicio. Este nuevo espacio de memoria (conteniendo la copia del texto) debe ser remitido al exterior de la función para su posterior utilización.

Ejercicio 3.9

Realizar una función cuyo prototipo sea

```
char * subcadena (char * p, unsigned int i, unsigned int n);
```

La función deberá retornar un puntero a un nuevo espacio de memoria dinámica conteniendo el fragmento del string *p* que va desde el carácter *i* tomando desde ahí *n* caracteres. Por ejemplo, el siguiente llamado a la función:

```
subcadena("Universidad Católica Argentina", 8, 5)
```

debería retornar el string "dad C".

Considerar que:

- el string resultante también debe tener ‘\0’ de finalización
- no puede modificar el string recibido como parámetro
- el valor de *i* puede estar más allá de los límites del string, en cuyo caso retornar NULL
- el valor de *n* puede ser mayor que la cantidad de caracteres que hay antes del final del string, en cuyo caso debería finalizar la copia cuando finaliza el string original.

Ejercicio 3.10

Dada la estructura:

```
typedef struct
{
    char * txt;
    unsigned int longitud;
}
t_texto;
```

hacer una función que permita ingresar un texto de longitud indefinida y guardarlo en una variable externa de este tipo, completando ambos campos. Para guardar el texto, utilizar la función programada en el *ejercicio 3.7* sin modificarla.

Ejercicio 3.11

Dada la estructura:

```
typedef struct
{
    char contenido;
    unsigned int descubierto;
}
t_casillero;
```

Programar el juego del buscaminas para un tablero de 10 x 10 casilleros. Cada variable de tipo `t_casillero` guarda el *contenido* del casillero ('*' para la bomba o los caracteres de '0' a '9' para indicar la cantidad de minas a su alrededor) y si el casillero fue *descubierto* o no (utilizando dicho campo como booleano).

Ejercicio 3.12

Dada la estructura:

```
typedef struct
{
    int a, b;
}
t_dosint;
```

Programar una función que intercambie los valores de los campos `a` y `b` de una variable de este tipo que es externa a la presente función. Dentro de esta función, utilizar la función programada en el *ejercicio 3.1* (sin modificarla).

Ejercicio 3.13

Dada la estructura:

```
typedef struct
{
    char * txt;
    unsigned int longitud;
}
t_texto;
```

hacer las siguientes funciones:

- a. Una función que me permita ingresar una cantidad indefinida de frases diferentes, las cuales se irán guardando en un arreglo dinámico de variables `t_texto`. Para cargar cada frase, utilizar la función del ***ejercicio 3.10***, sin modificarla. Utilizar una variable que contenga `NULL` en el campo `txt` como marca de finalización. La carga de frases finaliza cuando se ingresa un texto de tamaño cero.
- b. Una función que reciba como parámetro un arreglo dinámico de variables `t_texto` (el cual finaliza con la variable cuyo campo `txt` contiene `NULL`) y que lo ordene alfabéticamente.
- c. Una función que imprima una variable de tipo `t_texto` por pantalla (solo el texto).
- d. Una función que reciba como parámetro un arreglo dinámico de variables `t_texto` (el cual finaliza con la variable cuyo campo `txt` contiene `NULL`) y que lo imprima.

Implementar todas estas funciones en un programa donde se ingresen frases por teclado. Luego ordenar alfabéticamente dicho listado e imprimirlo.