



Práctica Dirigida 5

Punteros II

1. Declare un arreglo tridimensional, proceda a imprimir el arreglo utilizando punteros.
2. Utilizando doble indirección, diseñar un programa que calcule la traspuesta de la matriz:

$$A = \begin{bmatrix} 1 & -1 & 5 \\ 2 & 0 & 1 \\ -7 & 3 & 0 \end{bmatrix}$$

y se le sume a la matriz

$$B = \begin{bmatrix} 1 & 2 & 7 \\ 1 & 3 & 4 \\ 5 & -1 & 1 \end{bmatrix}$$

3. Utilizando punteros y un vector de tipo entero que tenga por nombre year, escribir un programa que genere el siguiente resultado:
El mes 1 tiene 31 días.
El mes 2 tiene 28 días.
.....
El mes 12 tiene 31 días
4. Los arreglos de punteros son útiles para declarar un arreglo de cadenas de caracteres. Si una cadena de caracteres puede escribirse como `char *c`; un arreglo de cadenas de caracteres podrá escribirse como `char **c`; o `char *c[x]`. Utilizando arreglo de punteros defina un arreglo de cadenas de caracteres conteniendo los 7 días de la semana. Implementa una función con el prototipo `void imprimirCadenas(char **str)`; que imprima las 7 cadenas.
5. Escribir una función que permita calcular el cubo de un número dado. Utilice el concepto de puntero a función.
6. Implementar una calculadora con las 4 operaciones básicas (+, -, *, /). Utilice un arreglo de punteros a funciones.
7. Modifique el ejercicio anterior y utilice un puntero a función como parámetro de otra función.

Cadena de Caracteres I

1. Escribir un programa que, al recibir como datos una cadena de caracteres y un caracter, determine cuántas veces se encuentra el caracter en la cadena. Utilice punteros y funciones.
2. Con el objetivo de poner en práctica el uso de funciones, arreglos, punteros y aritmética de punteros implemente funciones para:
 - i) Determinar la longitud de una cadena
 - ii) Realizar una copia de cadena
 - iii) Realizar la concatenación de cadenas.
 - iv) Realizar la Comparación de cadenas
3. Escriba la función `int busca(char s[], int c)` que regresa la posición de la primera aparición de `c` en la cadena `s` (es decir, `s[i] == c`) o `-1` si no está. Escriba la función `char* busca(char *s, int c)` que regresa un apuntador a la primera aparición de `c` en la cadena `s` o `NULL` si no está.
4. Escriba la función `void invierte(char s[])` que invierte la cadena `s`. Luego, escriba una versión con punteros `void invierte(char *s)`. Ejemplo: Entrada: roma. Salida: amor.
5. Escriba la función `void borra(char s[], int c)` que borra todas las apariciones de `c` en la cadena `s`. Escriba una versión con punteros `void borra(char *s, int c)`. Ejemplo: Entrada: catarata. Salida: ctrt.
6. Escriba las funciones `int binario(char *s)` y `int octal(char *s)` que conviertan una cadena escrita en binario u octal a un `int`.
7. Escriba una función `int hexadecimal(char *s)` que convierta una cadena escrita en hexadecimal a un `int`. Los dígitos hexadecimales del 10 al 15 se pueden representar con `abcdef` o con `ABCDEF`.
8. Escriba una función `void adecimal(char *s, int n)` que convierta `n` a una cadena escrita en decimal.
9. Escriba `int abinario(char *s, int n)` y `int aoctal(char *s, int n)` que conviertan `n` a una cadena escrita en binario u octal.
10. Escriba una función `void ahexadecimal(char *s, int n)` que convierta `n` a una cadena escrita en hexadecimal.
11. Escribir un programa que genere aleatoriamente una cadena de 50 caracteres formado por letras mayúsculas.
12. Considere el ejercicio anterior y determine la cantidad de veces que aparece cada letra.
13. Considere los dos ejercicios anteriores y reescriba la versión para letras minúsculas.
14. Escribir un programa que genere aleatoriamente 40 caracteres formadas por letras mayúsculas y minúsculas.
15. En el ejercicio anterior, agregar al final la cadena que resulta de intercambiar una letra minúscula por mayúscula y viceversa.

16. Escriba un programa que genere aleatoriamente una cadena de 40 caracteres formado solamente por letras y espacios en blanco. Muestre la cadena resultante.
17. Escriba una función que al recibir el nombre de una cadena, muestre dicha cadena en orden invertido.
18. genere una cadena formada por letras y espacios en blanco luego muestre la cadena generada y la cadena que surge de esta al eliminar los espacios en blanco. Muestre además la longitud de esta última.
19. Escriba un programa que lea una palabra de longitud no mayor a 20 y diga si la palabra es palíndroma o no. Ejemplo las siguientes palabras lo son: Ana , Madam, reconocer.