

Trabajo Práctico "Vectores y Punteros"

NOTA: Todos los ejercicios deben estar documentados siguiendo las reglas de trabajo de Doxygen. Deben tener un main con el cual se puedan probar las funciones solicitadas, como así también el Makefile correspondiente.

4.1

Escriba una función que imprima, byte a byte, los bytes que constituyen una variable long cuya dirección se recibe como argumento en la función.

4.2

Se ingresan 40 datos numéricos, de los cuales hay que mostrar la sumatoria de todos ellos, el promedio, la suma de los elementos que ocupan la posición par y el promedio de los valores positivos.

4.3

Escribir una función que reciba como argumento un puntero a char, la asuma como una secuencia en ASCII terminada en 0 (a los fines de la evaluación, este carácter en C es '\0') y devuelva la secuencia invertida.

void stringReverse (char *) ;

4.4

Escribir una función que reciba como argumento puntero a una secuencia de bytes, la asuma como una secuencia en ASCII terminada en 0 (a los fines de la evaluación, este carácter en C es '\0'), calcule su longitud y la retorne de acuerdo al siguiente prototipo:

int myStrlen (const char *) ;

4.5

Escribir una función que dados dos punteros a caracteres y un número "n" entero recibidos como parámetros, compare lexicográficamente los primeros "n" caracteres a partir de cada puntero, devolviendo un valor positivo, cero o negativo, si el primer string está primero alfabéticamente, si los dos strings están en el mismo orden alfabético o si el segundo está primero alfabéticamente correspondientemente. (Función strncmp() de la biblioteca <string.h >). El prototipo de la función pedida es:

int myStrncmp (const char *s1, const char *s2, int n) ;

4.6

Escribir una función que responda al siguiente prototipo:

void replace (char *s, char nuevo, char viejo) ;

y reemplace en la cadena "s" todas las apariciones del caracter "viejo" por el carácter "nuevo".

4.7

- Escribir 3 funciones que, recibiendo una cadena de caracteres como argumento, permitan determinar si la cadena es válida como dirección IP, como dirección de correo electrónico, y como número de tarjeta de crédito.
- Escribir una función denominada **validateString()** que recibiendo una cadena de caracteres y una función de validación (pasada por puntero), determine si la cadena es válida conforme al criterio de validación indicado, retornando en consecuencia true o false por su nombre.