

## PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática  
Facultad de Ciencias Exactas y Tecnología - UNT

### Trabajo Práctico N° 6

#### TEMA: Arreglos

“Un arreglo es una colección de variables ordenadas e indexadas, todas de idéntico tipo que se referencian usando un nombre común”.

#### Funciones de biblioteca del archivo de cabecera <string.h>

Las variables cad1 y cad2 son arreglos de caracteres.

<b>int strlen(cad1)</b>	Retorna la longitud de cad1	
<b>int strcmp(cad1, cad2)</b>	Compara cad1 con cad2, carácter a carácter	SI (cad1[i] < cad2 [i] ) ENTONCES Retorna entero < 0 SI (cad1[i] = cad2 [i] ) ENTONCES Retorna 0 SI (cad1[i] > cad2 [i] ) ENTONCES Retorna entero > 0
<b>int strncmp(cad1,cad2,n)</b>	Compara hasta n caracteres de la cad1 con cad2 sin diferenciar mayúsculas de minúscula.	SI (cad1[i] < cad2 [i] ) ENTONCES Retorna entero < 0 SI (cad1[i] = cad2 [i] ) ENTONCES Retorna 0 SI (cad1[i] > cad2 [i] ) ENTONCES Retorna entero > 0
<b>char *strcpy(cad1, cad2)</b>	Copia cad2 a cad1, incluyendo el terminador “\0”. Retorna cad1	
<b>char *strncpy(cad1, cad2,n)</b>	Copia hasta n caracteres de la cad2 a cad1. Retorna cad1. Rellena con “\0” si cad2 tiene menos de n caracteres	
<b>char *strcat(cad1, cad2)</b>	Concatena la cad2 al final de cad1. Retorna cad1	



Para el desarrollo del trabajo práctico, en todos los ejercicios, diseñar y escribir un algoritmo que resuelva la consigna propuesta. Luego, codificar en Lenguaje C. Realice pruebas para distintos conjuntos de datos y asigne convenientemente el tipo de los mismos.

## PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática  
Facultad de Ciencias Exactas y Tecnología - UNT

### 1. Arreglo con números

- Declare un arreglo de números enteros **arreNumero** y cargue con 20 números enteros aleatorios comprendidos en el intervalo [0,400]. Luego calcule y muestre por pantalla la suma de los valores que hay en posiciones pares del mismo (0,2,...) y cuente cuántos números pares hay en las posiciones impares (1,3,...).
- Modifique el programa anterior para que el número de valores aleatorios no sea fijo sino que se lea como entrada y sea como máximo 20. ¿Qué ocurre si indicamos más de 20? Lea también los valores para conformar el intervalo en consideración, realice los controles convenientes.

Para trabajar declare y defina las siguientes funciones:

```
void cargarArreglo(int arreglo[], int tama, int num1, int num2)
void mostrarArreglo(int arreglo[], int tama)
void sumarycontarArreglo(int arreglo[], int tama)
```

### 2. Arreglos con caracteres

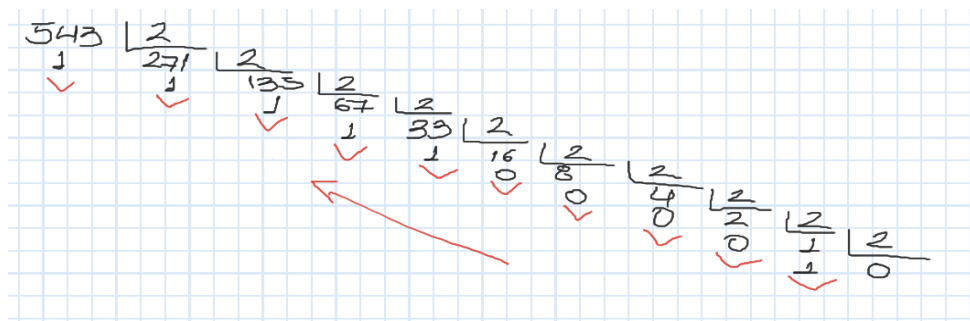
- Dado el nombre y el apellido de una persona, por separado, unirlos en una única cadena de manera que quede de la forma: "Apellido, Nombre". No use la función **strcat()**.
- A partir del nombre completo de una persona (al menos un apellido y dos nombres), compactar la frase en el mismo arreglo, es decir, eliminar los espacios en blanco. Además, sume los equivalentes en código ascii de las letras mayúsculas. Por ejemplo:

Entrada: "Héctor Eduardo Reglero Montaner"

Salida: "HéctorEduardoRegleroMontaner" - Suma: 300 (72+69+82+77)

### 3. Conversor de Números

El sistema de numeración binario utiliza sólo dos dígitos: 0,1 para representar números. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición del dígito menos uno. Para convertir un número decimal a base 2 se realizan divisiones sucesivas en dos de manera que los restos forman el equivalente en binario.



De esta forma  $(543)_{10} = (100001111)_2$

La tarea consiste en leer un número decimal y mostrar su equivalente binario. Luego de almacenar los restos, use un arreglo auxiliar para guardar correctamente el número en base 2.

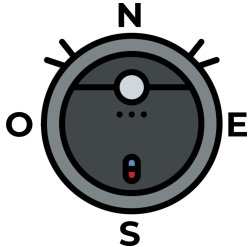
## PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática  
Facultad de Ciencias Exactas y Tecnología - UNT

---

### 4. Aspirabot

Se tiene una aspiradora robot que puede estar mirando en una de 4 direcciones posibles: norte, sur, este y oeste. A continuación se muestra una mirando hacia el norte, vista desde arriba.



Recibe una lista de instrucciones, de dos tipos:

- **Girar a la izquierda**, en sentido antihorario visto desde arriba:  
norte → oeste → sur → este → norte
- **Girar a la derecha**, en sentido horario visto desde arriba:  
norte → este → sur → oeste → norte

Comienza mirando hacia el norte. El objetivo final es que quede mirando al sur. Por ejemplo, si un robot que mira hacia el norte recibe una orden de girar a la derecha (en sentido horario), quedaría mirando hacia el este.

El robot va ejecutando todas las instrucciones una por una, y el proceso termina cuando el robot queda mirando al sur. No importa si luego existen más instrucciones.

Se desea conocer cuantas instrucciones hay que procesar hasta que el robot quede mirando al sur, podría nunca suceder. Escribir una función que calcule este dato a partir de la cadena de instrucciones.

Entrada: HAHAHHHAAHA - Salida: 8

Entrada: HAA - Salida: -1

**Nota:** Una H indica un giro en sentido horario, y una A un giro en sentido antihorario.

### 5. Adivinando la Palabra

Programar un prototipo para un juego llamado Adivinando la Palabra. El juego funciona así:

- Primero se debe ingresar una palabra. Luego según la cantidad de caracteres, se debe mostrar en pantalla guiones incógnita reemplazando los caracteres.
- Un usuario tiene que adivinar la palabra ingresando alguna letra.
  - Si la letra se encuentra en la palabra, se debe volver a mostrar en pantalla los guiones incógnita, salvo las letras encontradas por el usuario.
  - Si el usuario ingresa una letra que no está en la palabra se debe descontar un intento. Y tiene un máximo de 3 intentos para adivinar la palabra.

Si el participante gana se debe mostrar un mensaje informando que adivinó la palabra o si perdió, informar que agotó los tres intentos.

Ejemplo:

- 1- Ingresa palabra: **programacion**
- 2- Muestra en pantalla: - - - - -
- 3- Ingresa una letra: **a**
- 4- Muestra en pantalla: - - - - a - a - - - Intentos: **3**
- 5- Ingresa una letra: **e**
- 6- Muestra en pantalla: - - - - a - a - - - Intentos: **2**