

## 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	



Resuelva el punto 1 con detenimiento, en el mismo se busca de forma sencilla aprender a manejar el recorrido en un arreglo.

## PROGRAMACIÓN

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

---

### 1. Calentando Motores

Genere un arreglo de números enteros de tamaño fijo, con 15 elementos, y llénelo con números aleatorios en el rango [10, 350].

- Muestre todos los números almacenados en el arreglo.
- Recorra el arreglo para encontrar y mostrar el número más grande.
- Calcule y muestre el promedio de todos los números almacenados en las posiciones impares.
- Reemplace cada número mayor a 300 por un número que ingrese el usuario por teclado.
- Incremente en uno todos los números impares del arreglo.
- Modifique el programa para que el número de valores aleatorios no sea fijo sino que se lea como entrada y sea como máximo 15. ¿Qué ocurre si indicamos más de 15?

Para trabajar declare y defina las siguientes funciones:

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

**Nota:** num1 y num2 son los valores del rango en el que se debe cargar el arreglo

### 2. Verificando contraseñas

Escriba un programa en C que permita al usuario crear una contraseña y verificar si cumple con los siguientes requisitos:

Debe tener al menos 8 caracteres, contener al menos una letra mayúscula, una letra minúscula y un número. Además debe ingresar nuevamente la contraseña y corroborar si ambas son iguales.

El programa debe mostrar un mensaje indicando si la contraseña es válida o cuál de los requisitos no se cumplió. El proceso puede repetirse hasta que se ingrese una contraseña válida.

### 3. Concatenando cadenas

- 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, agregue un punto al final.

Por Ejemplo:

Entrada: "Héctor Eduardo Reglero Montaner"

Salida: "HéctorEduardoRegleroMontaner."

## PROGRAMACIÓN

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

---

### 4. Suma de Matrices

Escriba un programa en C que permita al usuario ingresar dos matrices cuadradas y calcular su suma. Las tres matrices deben mostrarse en pantalla. Realice las funciones para cargar una matriz, mostrar una matriz y sumar dos matrices.

Una matriz A es cuadrada si el número de filas es igual al número de columnas.

Siendo A y B matrices de igual dimensión, la suma de A y B se obtiene sumando los términos situados en el mismo lugar.

### 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, basándose en la longitud de la palabra ingresada, el programa muestra guiones incógnitos que representan cada carácter de la palabra.
- Un usuario tiene que adivinar la palabra ingresando alguna letra.
  - Si la letra está en la palabra, el programa actualiza los guiones incógnitos, revelando las letras adivinadas.
  - Si el jugador introduce una letra que no está en la palabra, se le resta un intento. El jugador tiene un límite de 3 intentos.

El juego continúa hasta que el jugador adivine la palabra o agote sus 3 intentos. Si el jugador adivina la palabra, el programa muestra un mensaje de felicitación. Si el jugador agota sus 3 intentos sin adivinar la palabra, el programa muestra un mensaje de derrota y revela la palabra correcta.

Ejemplo:

- 1- Jugador 1: Ingresar palabra: **programacion**
- 2- Muestra en pantalla: - - - - -
- 3- Jugador 2: Ingresar una letra: **a**
- 4- Muestra en pantalla: - - - - a - a - - - Intentos: **3**
- 5- Jugador 2: Ingresar una letra: **e** (No hay letra e en la palabra)
- 6- Muestra en pantalla: - - - - a - a - - - Intentos: **2**

### 6. Cazador de tesoros

Programe un juego con las siguientes reglas:

El juego se desarrolla en un mapa cuadrado, representado por una matriz, donde se oculta un tesoro en una ubicación aleatoria.

El objetivo del jugador es **adivinar las coordenadas (X, Y)** en dónde se encuentra el tesoro en el mapa y para ello **cuenta con cinco intentos** para encontrarlo.

Cada celda del mapa es inicialmente desconocida. El jugador puede explorar una celda ingresando coordenadas y el programa marcará esa celda como explorada.

## PROGRAMACIÓN

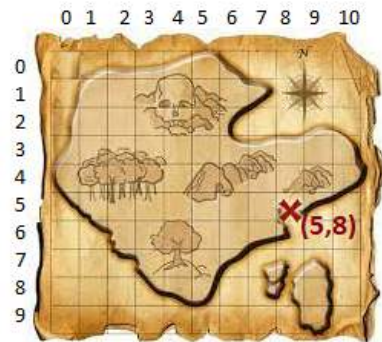
Programador Universitario - Licenciatura en Informática - Ingeniería en Informática  
Facultad de Ciencias Exactas y Tecnología - UNT

---

El programa calculará y mostrará la distancia total desde las coordenadas ingresadas por el jugador hasta la ubicación del tesoro después de cada intento.

Si el jugador adivina las coordenadas del tesoro, el programa mostrará un mensaje de felicitación y el número de intentos que le tomó al jugador para encontrar el tesoro.

Si el jugador agota sus intentos sin encontrar el tesoro, el programa mostrará un mensaje de derrota y revelará las coordenadas del tesoro.



### Implementación

El programa utiliza una matriz bidimensional para representar el mapa y las ubicaciones que serán exploradas por el jugador.

Las celdas no exploradas se representan con " . ", las exploradas con "O". La ubicación del tesoro se mantiene oculta hasta finalizar y se mostrará con "X".

Luego de cada selección de coordenadas debe mostrar nuevamente el mapa con los puntos seleccionados.

Ejemplo: Si el tesoro se encuentra en la posición (5,8) y el jugador supone que está en la posición (3,9), el programa debe indicar que se encuentra a 3 celdas de distancia y descontar un intento.

Realice al menos una función para inicializar el mapa y una función para mostrarlo.

## PROGRAMACIÓN

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

### EJERCICIOS EXTRAS

#### 1. Misión: Rover Planetario

Te encuentras en una misión de exploración en un planeta previamente mapeado.

El objetivo es recorrer un sector de terreno representado por un **mapa de 10x10** casilleros.

Operarás un Rover que ingresará en la **posición (0,0)** del mapa.

Dentro del terreno, se han colocado **tres objetos no identificados en ubicaciones aleatorias**.

Tu tarea es pilotar el Rover para recorrer el mapa y recoger los objetos al pasar por las casillas correspondientes.

Sin embargo, hay restricciones importantes:

El Rover tiene una fuente limitada de energía, lo que significa que no puede pasar por una misma casilla dos veces. Además, el Rover no puede salir del mapa. Si sucede alguna de las dos, la misión se considerará un fracaso y terminará.

El éxito de la misión se logra cuando recuperas los tres objetos no identificados.

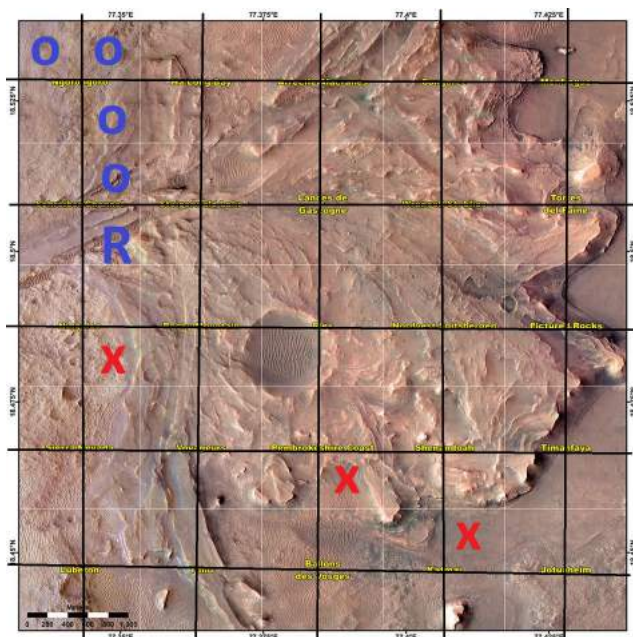
Tus acciones serán dar instrucciones al Rover para moverlo por el mapa. Cada movimiento estará codificado de la siguiente manera:

- 'N' para moverte hacia el norte.
- 'S' para moverte hacia el sur.
- 'E' para moverte hacia el este.
- 'O' para moverte hacia el oeste.

A medida que el Rover se mueve, debes mantener un registro de todas las instrucciones dadas y actualizar el mapa para reflejar los cambios en el terreno explorado.

Las casillas no exploradas se representarán con "-", los objetos no identificados con "X", las casillas exploradas con "O" y la casilla en donde se encuentra el Rover con "R".

Al finalizar tienes que mostrar toda la secuencia de instrucciones completa desde que partió de la posición (0,0).



Rover partió de (0,0)

Secuencia parcial: **ESSS**

Rover se encuentra en (3,1)

Los objetos extraños se encuentran en:

**(5,1),(7,5),(8,7)**

Misión en curso.

Mapa a mostrar:

O	O	-	-	-	-	-	-	-	-
-	O	-	-	-	-	-	-	-	-
-	O	-	-	-	-	-	-	-	-
-	R	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-
-	X	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	X	-	-	-	-
-	-	-	-	-	-	-	X	-	-
-	-	-	-	-	-	-	-	-	-