

# Universidad Nacional de Asunción - Facultad de Ingeniería

Computación – 2021c1

Primer Examen Final - Tema 2

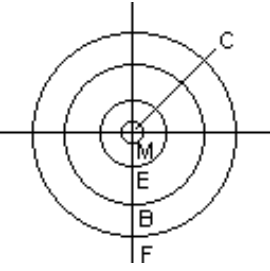
Fecha: 26/06/2021

**TEMA 2:** Puede ser realizado en DevC++ o similar • El código deberá ser copiado o alzado al VPL (uno de los 2, NO ambos). Si elige alzar, solo puede hacerlo una vez, se recomienda alzar cuando ya no modificará el tema. No se contarán con casos de prueba • Será corregido manualmente por los docentes de la cátedra • Valor 35% del total de puntos. El nombre del archivo debe ser T2F1\_numeroCI seguido de la extensión .cpp.

**Ejemplo** T2F1\_1234567.cpp

## Stand de Tiro S.A.

Un tirador efectúa 3 disparos a un papel de tiro. Este tiene 5 zonas con la denominación y el puntaje asociado para cada disparo. Los detalles se muestran en la **Figura**:

ZONA	DENOMINACION	PUNTAJE		<b>FORMULA</b> $R = \sqrt{a^2 + b^2}$
$0 \leq R < 2$	C=CENTRO	10		
$2 \leq R < 4$	M=MEDIA	8		
$4 \leq R < 6$	E=EXTERNA	5		
$6 \leq R \leq 10$	B=BORDE	1		
$R > 10$	F=FUERA	0		

1. Se pide crear las siguientes estructuras de datos:

**Puntaje** (denominación de 7 caracteres, puntaje short sin signo)

**Punto** (a y b, short sin signo)

**Tirador** (nombre de 20 caracteres, disparos vector para 3 disparos que guardarán los puntos)

2. Crear una aplicación que muestre el siguiente menú:

--.-- Disparos al blanco --.--

1. Leer datos de Tirador

2. Salir

Ingrese la Opcion->

Al seleccionar la opción 1 se debe solicitar al usuario que ingrese el nombre y apellido del Jugador (en una sola variable) y generar aleatoriamente tres pares de números que equivalen a las coordenadas de los puntos del plano cartesiano correspondientes a cada disparo (par de números (a, b) comprendidos entre -10 y 10, inclusive).

**Obs.:** se hizo coincidir el centro del papel de tiro con el centro del plano cartesiano.

El programa debe calcular el mejor puntaje del tirador, de los 3 disparos, de acuerdo al valor de **R**. La distancia **R** del punto P(a, b) al centro (0, 0) se calculará por medio de la fórmula expresada en el cuadro de arriba. La denominación y el puntaje debe cargarse en la estructura **Puntaje**.

**Se deben imprimir:** el nombre del jugador, sus 3 disparos y el mejor disparo (la denominación con su puntaje).

Ejemplos de Salida:

```
--.-- Disparos al blanco --.--
1. Leer datos de Tirador
2. Salir
Seleccione la Opcion-> 1
Ingrese el nombre del Tirador: Raul Pesoa
Raul Pesoa
Disparos:
10,10
7,18
7,0
Mejor tiro: BORDE Puntaje: 1
```

```
--.-- Disparos al blanco --.--
1. Leer datos de Tirador
2. Salir
Seleccione la Opcion-> 1
Ingrese el nombre del Tirador: Carlos Romero
Carlos Romero
Disparos:
0,1
15,13
12,3
Mejor tiro: CENTRO Puntaje: 10
--.-- Disparos al blanco --.--
1. Leer datos de Tirador
2. Salir
Seleccione la Opcion-> 2
-----
Process exited after 308.5 seconds with return
Presione una tecla para continuar . . .
```

**Todas las operaciones deben realizarse mediante funciones:**

- Mostrar el menú y leer la opción
- Leer nombre y generar disparos
- Generar número aleatorio para a y b
- Determinar el mejor puntaje
- Calcular los valores R para cada disparo
- Puntaje y denominación del mejor disparo

En la función **main NO** deben realizarse los procesos, solo debe invocarse a las funciones que realizan esos procesos.

**Los siguientes puntos serán evaluados, con 4 puntos cada uno, totalizando 40 puntos:**

- |                                       |   |
|---------------------------------------|---|
| 1. Definición de las estructuras      | 7. Proceso a través de funciones                  |
| 2. Presentación del Menú              | 8. Estructura de la función main                  |
| 3. Leer Nombre y Apellido del tirador | 9. Referencia a las estructuras en los parámetros |
| 4. Generar números aleatorios         | 10. Impresión de Resultados                       |
| 5. Calcular valores R                 |   |
| 6. Determinar mejor puntaje           |   |

**Universidad Nacional de Asunción - Facultad de Ingeniería**  
**Cátedra de Computación – 1er Examen Final – 26/06/2021 – Tema 3**

---

Para la transmisión segura de mensajes, estamos interesados en implementar una función que permita codificar mensajes de texto de hasta 100 caracteres de longitud (sin considerar el carácter de finalización de cadena). Para ello, debemos definir la función `codificar()`, que recibe dos cadenas:

- Una contiene el mensaje original (que puede contener espacios, tal como lo muestra el ejemplo).
- En la otra deberá almacenarse el mensaje codificado.

La correcta declaración de la función `codificar()` vale (1p). Esta función implementa el siguiente método de codificación:

- a) Se introduce (por teclado) una clave de 10 dígitos distintos (entre 0 y 9). *(Puede suponerse que el usuario siempre cargará una clave conformada por 10 dígitos distintos -entre 0 y 9-, pero debe validarse que estos dígitos no estén en orden ascendente)* (2p)
- b) Si el texto a codificar tiene menos de 100 caracteres, se completa al final con tantos asteriscos como sea necesario para completar los 100 caracteres. (1p)
- c) El texto resultante se coloca en una matriz de 10 filas y 10 columnas, de modo que cada carácter ocupe un elemento de la matriz, fila por fila, de izquierda a derecha y de arriba hacia abajo *(ver el ejemplo)*. (3p)
- d) Se toma el primer dígito de la clave, y la columna correspondiente a ese dígito se pasa al mensaje codificado. Se toma el segundo dígito de la clave, y la columna correspondiente a ese dígito se agrega al final del mensaje codificado. Se repite este procedimiento hasta utilizar todos los dígitos de la clave. (5p)
- e) Se agrega al final del mensaje codificado la clave utilizada para su codificación. (2p)

**Observaciones:**

- Se debe definir necesariamente la función `codificar()` que realice lo solicitado. No se aceptará que la funcionalidad se desarrolle en el `main()`.
- Se proporciona un archivo con código inicial. Por practicidad en el proceso de desarrollo del ejercicio, en el `main()` se encuentra *pre-cargado* el mensaje empleado para el ejemplo. De esta manera, solo se debe llamar a la función `codificar()` con este mensaje y ver el mensaje codificado. Sin embargo, `codificar()` deberá funcionar (de acuerdo con lo especificado anteriormente) para cualquier caso de entrada.

**Ejemplo**

Mensaje original: “LA CRIPTOGRAFIA ES LA CIENCIA DE CIFRAR Y DESCIFRAR MENSAJES USANDO TECNICAS MATEMATICAS”

Clave generada: 8204975613

La matriz es:

L	A		C	R	I	P	T	O	G
R	A	F	I	A		E	S		L
A		C	I	E	N	C	I	A	
D	E		C	I	F	R	A	R	
Y		D	E	S	C	I	F	R	A
R		M	E	N	S	A	J	E	S
	U	S	A	N	D	O		T	E
C	N	I	C	A	S		M	A	T
E	M	A	T	I	C	A	S	*	*
*	*	*	*	*	*	*	*	*	*

El mensaje codificado es:

O ARRETA\*\* FC DMSIA\*LRADYR CE\*RAEISNNAI\*GL ASET\*\*TSIAFJ MS\*I NFCSDSC\*PECRIA  
O A\*AA E UNM\*CIICEEACT\*8204975613