

### Indicaciones específicas:

- Esta evaluación contiene 10 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
  - p1.cpp
  - p2.cpp
  - p3.cpp
- Deberás subir estos archivos directamente a [www.gradescope.com](http://www.gradescope.com), uno en cada ejercicio.
- En los archivos .cpp evitar incluir tildes en la impresión de textos y comentarios.

### Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
  - Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)
  - Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)
  - Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
  - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
  - Capacidad de aplicar conocimientos de ingeniería(nivel 2)
  - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

## Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	7	
2	6	
3	7	
Total:	20	

1. (7 points) **Diamante brillante**

**Tema a evaluar:** Estructuras de control

La empresa "**Diamante brillante**" quiere realizar una campaña de marketing realizando un desafío de programación en sus redes sociales para sus consumidores. El objetivo del desafío es crear el logo de su marca. Usted desea participar usando una solución en el lenguaje C++ para generar el logo que se muestra a continuación:

Listing 1: Ejemplo del logo

```
  **
 *  *
*    *
*    *
 *  *
  **
```

Para resolver el ejercicio debe utilizar **estructuras de control selectivas e iterativas**, donde el input y output se obtiene de la siguiente forma:

**Input**

- Se recibe del usuario el tamaño del alto de la figura con un entero entre 4 y 50. Además, el programa debe validar que el valor ingresado es par.
- La figura tiene el mismo ancho y alto.

**Output**

- Se imprime todos los lados de la figura utilizando el caracter asterisco ("\*").
- Cada lado debe tener la mitad de caracteres asterisco ingresado por el usuario. Por ejemplo si ingresa el alto de 6, debe tener 3 asterisco por lado.

**IMPORTANTE:** En este ejercicio no se permite el uso de bibliotecas.

A continuación se muestra algunos ejemplos de la ejecución correcta del código:

Listing 2: Ejemplo de resultado 1

```
Ingrese el alto:10

  **
 *  *
*    *
*    *
 *  *
  **

  **
 *  *
*    *
*    *
 *  *
  **

  **
 *  *
*    *
*    *
 *  *
  **

  **
 *  *
*    *
*    *
 *  *
  **
```

Listing 3: Ejemplo de resultado 2

```
Ingrese el alto:9
Ingrese el alto:11
Ingrese el alto:12
```

A circular arrangement of 24 asterisks. There are 2 asterisks at the top, 2 at the bottom, and 20 asterisks arranged in a ring around the perimeter.

Listing 4: Ejemplo de resultado 3

```
Ingrese el alto:200
Ingrese el alto:20
```

A circular arrangement of 24 asterisks (\*) forming a ring. The asterisks are positioned at regular intervals around the circle, with two at the top and bottom, and the rest distributed evenly along the sides.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	La ejecución es correcta (1pts).	La ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente (1pts).	El código no está optimizado y la ejecución es deficiente (0pts).

**2. (6 points) Suma de números Binarios**

**Tema a evaluar:** Funciones (nivel básico) y recursividad

Crear un programa que realice la conversión de números decimales a **números Binarios**. El programa debe solicitar un número repetidamente y mostrar la conversión del número a binario hasta que el usuario ingrese el valor cero (0). Finalmente debe mostrar los números binarios ingresado y la suma de estos números.

Debe implementar al menos 2 funciones realizando el pase por valor de las variables:

1. Realizar la conversión de un número decimal a binario de forma recursiva.
2. Imprimir los números binarios y la suma de estos. Para realizar la suma, se sugiere sumar los números decimales y convertir el número a binario.

Además, para convertir el número a binario debe implementar una función recursiva, de la siguiente forma:

$$BINARIO(A) = \begin{cases} String(A), & \text{si } A \leq 2 \\ BINARIO(A/2) + String(A\%2), & \text{si } A \geq 2 \end{cases} \quad (1)$$

**IMPORTANTE:** En este ejercicio solo se permite el uso de la biblioteca String.

A continuación se muestra algunos ejemplos de la ejecución correcta del código:

Listing 5: Ejemplo de resultado 1

```
----
NUMEROS BINARIOS
----
INGRESE NUMERO:12
El numero 12 en binario es: 1100
----
NUMEROS BINARIOS
----
INGRESE NUMERO:25
El numero 25 en binario es: 11001
----
NUMEROS BINARIOS
----
INGRESE NUMERO:-6
Solo se realiza el calculo para numeros positivos.
----
NUMEROS BINARIOS
----
INGRESE NUMERO:0
La suma de los numeros binarios 1100,11001, es 100101.
```

Listing 6: Ejemplo de resultado 2

```

----
NUMEROS BINARIOS
----
INGRESE NUMERO:300
El numero 300 en binario es: 100101100
----
NUMEROS BINARIOS
----
INGRESE NUMERO:40
El numero 40 en binario es: 101000
----
NUMEROS BINARIOS
----
INGRESE NUMERO:0
La suma de los numeros binarios 100101100,101000, es
    101010100

```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (2pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (1.5pts)	La ejecución es correcta (1pts).	La ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente (1pts).	El código no está optimizado y la ejecución es deficiente (0pts).

### 3. (7 points) Plan de contingencia en Unidad Renal

**Tema a evaluar:** Funciones (nivel intermedio) y punteros

Usted se encuentra trabajando en el área de Tecnologías de Información de un Hospital y han experimentado un ataque de ransomware que ha comprometido a los computadores y servidores con Windows.

Debido a que no se tiene certeza cuando se van a reestablecer los sistemas, se decide activar el **plan de contingencia**. Los doctores de la Renal se están quejando por cálculos incorrectos de la función renal que es realizado de forma manual. Por tal motivo su equipo decide implementar un pequeño programa en Linux para ayudar a los usuarios. Se le solicita crear un programa que realice el cálculo con la siguiente fórmula:

- Si el paciente es Hombre:

$$FUNCION\_RENAL(mL/min) = \frac{(140 - edad) * peso(Kg)}{(72 * Creatinina(mg/dL))} \quad (2)$$

- Si la paciente es Mujer:

$$FUNCION\_RENAL(mL/min) = \frac{(140 - edad) * peso(Kg)}{(72 * Creatinina(mg/dL))} * 0.85 \quad (3)$$

Además, le solicita que preserve la seguridad de los datos de los pacientes creando una opción para eliminar los datos.

El programa debe iniciar con un menú para que el usuario realice las siguientes opciones:

1. Registrar datos de paciente: Solicitar los siguientes datos:

- ID de paciente: String.
- Sexo: Char.
- Edad: Integer.
- Peso: Float.
- Creatinina: Double.

2. Realizar cálculo de función renal: El programa debe realizar el cálculo y mostrarlo en pantalla, en caso no existan datos debe reportar que no puede realizar el cálculo.

3. Eliminar datos de paciente: Para eliminar los datos debe colocar los valores numéricos en 0 y los valores de texto en vacío.

4. Finalizar programa.

Para realizar el programa se le solicita realizar las opciones 1, 2 y 3 en funciones que reciban punteros de los datos del paciente.

**IMPORTANTE:** En este ejercicio solo se permite el uso de la biblioteca String.

A continuación se muestra algunos ejemplos de la ejecución correcta del código:



## Listing 7: Ejemplo de resultado 1

```
-----
MENU
-----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE FUNCION RENAL
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4): 1
1. REGISTRAR ID DE PACIENTE:P00085
2. REGISTRAR SEXO DEL PACIENTE (Hombre: H o h / Mujer: M o m
):h
3. REGISTRAR EDAD:30
4. REGISTRAR PESO (KG):85
5. REGISTRAR CREATININA (mg/dL):1.3
-----
MENU
-----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE FUNCION RENAL
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4): 2
EL CALCULO DE LA FUNCION RENAL DEL PACIENTE P00085 DE SEXO
HOMBRE ES: 99.8932 C-G(mL/min)
-----
MENU
-----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE FUNCION RENAL
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4): 3
LOS DATOS DEL PACIENTE FUERON ELIMINADOS.
-----
MENU
-----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE FUNCION RENAL
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4):
4
```

Listing 8: Ejemplo de resultado 2

```

----
MENU
----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE FUNCION RENAL
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4): 2
PRIMERO DEBE REGISTRAR LOS DATOS DEL PACIENTE.

```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	La ejecución es correcta (1pts).	La ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente (1pts).	El código no está optimizado y la ejecución es deficiente (0pts).