

## Indicaciones específicas:

- Esta evaluación contiene 09 páginas (incluye esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Debe crear un único proyecto CLION con archivos independientes por pregunta:
  - Pregunta1 → p1.h y p1.cpp
  - Pregunta2 → p2.h y p2.cpp
  - Pregunta3 → p3.h y p3.cpp
  - El archivo main.cpp debe quedar así:

```
#include "p1.h"
#include "p2.h"
#include "p3.h"
int main() {
    pregunta1();
    pregunta2();
    pregunta3();
    return 0;
}
```

- Deberás subir un archivo ZIP que contenga únicamente los archivos \*.h y \*.cpp directamente a [www.gradescope.com](http://www.gradescope.com)

## 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 (solo para uso del profesor)

Question	Points	Score
<a href="#"><u>1</u></a>	7	
<a href="#"><u>2</u></a>	6	
<a href="#"><u>3</u></a>	7	
Total:	20	

## 1. (7 points) Calificación de Sustentación de Tesis

Desarrolle un programa que permita ingresar calificaciones **cualitativas** (A=Excelente, B=Bueno, C=Regular, D=Necesita mejorar) para cuatro criterios de evaluación de una sustentación de tesis (Introducción, Pregunta Problema, Desarrollo de la Investigación y Conclusiones), calcule la calificación **cuantitativa** y finalmente dictaminar la **calificación final** de dicha sustentación (Desaprobado, Aprobado, Notable, Sobresaliente, Summa Cum Laude).

Características para considerar en el desarrollo del programa:

- El programa deberá validar que cada criterio de evaluación sea calificado con A, a, B, b, C, c, D o d
- Para calcular la **calificación cuantitativa** se tiene las siguientes reglas:
  - Una evaluación cualitativa A equivale a 5 puntos
  - Una evaluación cualitativa B equivale a 4 puntos
  - Una evaluación cualitativa C equivale a 3 puntos
  - Una evaluación cualitativa D equivale a 2 puntos
- Para calcular la **calificación final** en base a la calificación cuantitativa se tiene las siguientes reglas:
  - Menor a 13 → Desaprobado
  - 13 a 14 → Aprobado
  - 15 a 16 → Notable
  - 17 a 18 → Sobresaliente
  - 19 a 20 → Summa Cum Laude
- El programa deberá calcular y mostrar la calificación cuantitativa y la calificación final.
- El programa deberá ser resuelto usando como mínimo funciones para:
  - Ingreso y Validación de datos
  - Cálculo del promedio
  - Presentación de los resultados
- Usted debe decidir cuando las funciones requieren el uso de parámetros por valor y cuando por referencia.

Algunos ejemplos de dialogo de este programa serían:

Listing 1: Ejemplo 1

Ingreso de datos  
Introducción: X  
Introducción: 1  
Introducción: A  
Pregunta Problema: E  
Pregunta Problema: B  
Desarrollo de la Investigación: 5  
Desarrollo de la Investigación: C  
Conclusiones: F  
Conclusiones: B

Resultado  
Calificación Cuantitativa: 16  
Calificación Final: Notable

Listing 2: Ejemplo 2

Ingreso de datos  
Introducción: a  
Pregunta Problema: b  
Desarrollo de la Investigación: c  
Conclusiones: d

Resultado  
Calificación Cuantitativa: 14  
Calificación Final: Aprobado

Listing 3: Ejemplo 3

Ingreso de datos  
Introducción: c  
Pregunta Problema: c  
Desarrollo de la Investigación: d  
Conclusiones: d

Resultado  
Calificación Cuantitativa: 10  
Calificación Final: Desaprobado

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).
Optimización	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) Potencia de un número

Escribir un programa que contenga, entre otras funciones, una recursiva cuyo nombre sea **potencia** el cual permita calcular la potencia de un numero

Características para considerar en el desarrollo del programa:

- El programa deberá permitir ingresar la base y el exponente como números enteros
- Se debe validar que el exponente sea un numero entero mayor o igual a 0.
- La función **potencia** debe recibir la base y el exponente y debe retornar la potencia.
- El programa deberá ser resuelto usando como mínimo funciones para:
  - o Ingreso y Validación de datos usando parámetros de tipo punteros
  - o Cálculo de la potencia (función recursiva). Aquí usted decide el uso de parámetros por valor, referencia o puntero.
  - o Presentación de los resultados. Aquí usted decide el uso de parámetros por valor, referencia o puntero.

Algunos ejemplos de dialogo de este programa serían:

Listing 4: Ejemplo 1

```
Base: 2
Exponente: -2
Exponente: 3
Potencia: 8
```

Listing 5: Ejemplo 2

```
Base: 3
Exponente:0
Potencia: 1
```

Listing 6: Ejemplo 3

```
Base: 10
Exponente:1
Potencia: 10
```

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).
Optimización	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).

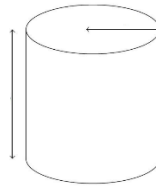
### 3. (7 points) Costo de construcción de una piscina circular

Elabore un programa para calcular el costo de adecuación de un terreno para la construcción de una piscina circular conociendo la siguiente información:

- El costo de excavación es de S/ 100 por  $m^3$
- El costo de recubrimiento es de S/ 150 por  $m^2$

Características para considerar en el desarrollo del programa:

- El programa deberá solicitar se ingrese las medidas (metros) de profundidad y radio de la piscina. Estos datos deben ser mayores a 0.



- Para todo efecto considere el valor de  $\pi = 3.14$
- El programa deberá calcular y mostrar los costos de excavación, costos de recubrimiento y costo total.
- El programa deberá ser resuelto usando como mínimo funciones para:
  - o Ingreso y Validación de datos usando parámetros de tipo puntero
  - o Cálculo de los costos. Aquí usted decide el uso de parámetros por valor, referencia o puntero.
  - o Presentación de los resultados. Aquí usted decide el uso de parámetros por valor, referencia o puntero.

Algunos ejemplos de dialogo de este programa serían:

#### Listing 8: Ejemplo 1

Construccion de Piscina Circular

Profundidad: 0

Profundidad: 2

Radio: -1

Radio: 5

Costos (s/.)

Excavacion: 15700

Recubrimiento: 21195

Total: 36895



## Listing 9: Ejemplo 1

Construccion de Piscina  
 Profundidad: 1.8  
 Radio: 9.8

Costos (s/.)  
 Excavacion: 54281.8  
 Recubrimiento: 61851.7  
 Total: 116134

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).
Optimización	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).