



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 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 únicamente los archivos *.h y *.cpp directamente a 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
<u>1</u>	7	
<u>2</u>	6	
<u>3</u>	7	
Total:	20	

1. (7 points) Sistema de evaluación cualitativa

Desarrolle un programa que permita ingresar las notas de 5 evaluaciones **cualitativas** (A=Excelente, B=Bueno, C=Regular, D=Necesita mejorar) y calcule el promedio final **cuantitativo** considerando únicamente las 4 notas cualitativas más altas.

Características para considerar en el desarrollo del programa:

- El programa deberá validar que las evaluaciones cualitativas sean A,B,C,D,a,b,c,d
- Para calcular el promedio final cuantitativo 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
- El programa deberá mostrar el promedio final cuantitativo, la condición de aprobado (promedio ≥ 13) o desaprobado (promedio < 13) y la evaluación cualitativa que fue eliminada.
- 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
Evaluación Cualitativa .1 : Z
Evaluación Cualitativa .1 : A
Evaluación Cualitativa .2 : D
Evaluación Cualitativa .3 : B
Evaluación Cualitativa .4 : C
Evaluación Cualitativa .5 : A
```

```
Resultado
Promedio : 17
Condición: Aprobado
Se eliminó: D
```

Listing 2: Ejemplo 2

Ingreso de datos
Evaluación Cualitativa .1 : d
Evaluación Cualitativa .2 : c
Evaluación Cualitativa .3 : c
Evaluación Cualitativa .4 : d
Evaluación Cualitativa .5 : d

Resultado
Promedio : 10
Condición: Desaprobado
Se eliminó: D

Listing 3: Ejemplo 3

Ingreso de datos
Evaluación Cualitativa .1 : A
Evaluación Cualitativa .2 : B
Evaluación Cualitativa .3 : C
Evaluación Cualitativa .4 : A
Evaluación Cualitativa .5 : B

Resultado
Promedio : 18
Condición: Aprobado
Se eliminó: C

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

Elabore un programa para calcular el costo de adecuación de un terreno para la construcción de una piscina rectangular 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, largo y ancho de la piscina. Estos datos deben ser mayores a 0.
- 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 7: Ejemplo 1

```
Construccion de Piscina
Profundidad: 0
Profundidad: 2.5
Largo: -1
Largo: 0
Largo: 10
Ancho: -3
Ancho: 3

Costos (s/.)
Excavacion: 7500
Recubrimiento: 14250
Total: 21750
```


Listing 8: Ejemplo 2

```

Construccion de Piscina
Profundidad: 2
Largo: 8
Ancho: 3

Costos (s/.)
Excavacion: 4800
Recubrimiento: 10200
Total: 15000

```

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).