

Programación III

1° Parcial

Instrucciones:

- Ingresar al Campus Virtual y descargar plantilla del repositorio indicado por el docente.
- Entrega del parcial: Commit y push de los ejercicios. Podrán hacer todos los commits y push que deseen. El que se corregirá es el último commit hasta el horario de entrega.
- En todos los ejercicios implementar un main que **al menos arroje por pantalla el resultado**.
- Exámen calculado para ser resuelto en 2 horas. Máximo 3hs.
- Puntos del examen: 10
- Cantidad mínima de puntos para aprobar: 6 (60%)

Consideraciones al momento de evaluar y en orden de importancia:

- Que la lógica de programación usada considere estrechamente el uso de las estructuras de datos enseñadas en clases y que las estructuras elegidas sean las más convenientes para resolver el problema. La Eficacia será considerada. La eficiencia no, al menos en esta instancia.
- El cumplimiento de lo especificado en el enunciado del ejercicio. El enunciado especifica lo mínimo a alcanzar para obtener el máximo puntaje.
- La consideración de los casos extremos y su abordaje con excepciones. Usar directivas Try, Throw y Catch.
- Tiempo de entrega.
- Compilación correcta de los programas.
- Ejecución del programa.

Restricción:

- No pueden cambiarse los parámetros de la función que se solicita en el enunciado. Es solicitud del cliente y debe respetarse.
- En aquellos ejercicios en los que se solicite la implementación de un método en una clase no puede resolverse usando los métodos ya implementados. Debe crearlo desde cero.

Ejercicio 1

Implemente una función **recursiva** que reciba como parámetros un **array** de tipo **entero** (Con signo y sin signo) y devuelva el dato más chico que hay en el array. (3 puntos)

Ejercicio 2

Implementar un método en la **clase lista** que reciba como parámetro un valor entero "pos". El método debe buscar el nodo en la posición "pos" y moverlo al inicio. Considerar el uso **excepciones de ser necesario**. (**Solo moviendo enlaces. No creando nodos nuevos**). (3 puntos)

Ejercicio 3

Agregue un método a la **clase Cola** que inserte un nodo con un nuevo dato después de la **enésima ocurrencia** de ese mismo dato partiendo la búsqueda **desde el tope**. Si no hay ocurrencias de mencionado dato, encolarlo al fondo. Considerar el uso **excepciones de ser necesario**. (4 puntos)

Nombre de Alumno:

Fecha: 15-09-2022

Clave:

Usuario de gitHub: