

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMIA**

**3ra. práctica (tipo B)**  
**(Primer Semestre 2023)**

Duración: 2h 50 min.

- En cada función el alumno deberá incluir, a modo de comentario, la estrategia o forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear STL, Plantillas o funciones no vistas en los cursos de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado. El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- **Solo esta permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías **iostream**, **omanip**, **cmath**, **fstream** y **cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_LAB3_P#` (donde # representa el número de la pregunta a resolver)

---

**Pregunta 1 (10 puntos)**

Luego de la derrota del ejército de Bowser en el examen parcial de Algoritmia. Bowser se ha preparado mejor y utilizando 2 robots Kupa, les ha encomendado la tarea de crear nuevos guerreros más potentes para enfrentar sus nuevas batallas. Cada robot Kupa han ido apilando los guerreros que han fabricado en una pila cada uno. Los guerreros que han creado tienen diferentes niveles de ataque, los cuales no necesariamente están en orden. A continuación, se muestra un ejemplo de las pilas de guerreros creados por ambos robots Kupa.

Robot Kupa 1: 85 -> 15 -> 56 -> 30 -> 27  
Robot Kupa 2: 45 -> 22 -> 68 -> 20

En un momento, Bowser solicita a los robots Kupa que les envíe los guerreros que han fabricado hasta ese momento, para ello, los robots Kupa deben unir ambas pilas en una sola. Recuerde que para esta tarea no debe crear una nueva pila, sino que debe emplear una de las 2 que ya tiene generadas.

Luego debe ordenar esta pila por nivel de ataque de mayor a menor, esto es por una estrategia establecida por Bowser. Aquí se muestra un ejemplo de la pila única ordenada, con los guerreros creados por ambos robots Kupa.

Pila única ordenada: 85 -> 68 -> 56 -> 45 -> 30 -> 27 -> 22 -> 20 -> 15

Finalmente, para pasar del mundo Kupa al mundo Champiñón, se debe realizar el paso de los guerreros de una pila a otra. Los guerreros deben pasar en el mismo orden.

Pila en el mundo champiñón: 85 -> 68 -> 56 -> 45 -> 30 -> 27 -> 22 -> 20 -> 15

Se le pide que desarrolle lo siguiente:

- Desarrolle una función que permita unir las dos pilas en una pila única. Para ello no puede utilizar memoria extra ni ninguna estructura, ni TADs adicionales (2 puntos).
- Desarrolle una función que permita ordenar la pila única, que contiene a todos los guerreros, por nivel de ataque de mayor a menor. En esta función no puede utilizar memoria extra y solo podrá utilizar una pila auxiliar, si lo considera necesario. (4 puntos).
- Desarrolle una función que permita pasar los guerreros de la pila única del mundo Bowser a la pila única del mundo champiñón, respetando el orden de los guerreros. Para esto no puede utilizar memoria extra ni ninguna estructura, ni TAD's adicional. (4 puntos).

## Pregunta 2 (10 puntos)

Como se mencionó en la evaluación pasada, un Equipo de Gestión de Problemas de Seguridad de la Información (EGPSI) tiene como función realizar el análisis, investigación y resolución de Problemas de Seguridad de la Información (PSI) que son consecuencia de incidentes repetitivos en la empresa o que hayan sido graves. El EGPSI recibe 5 listas, una por cada categoría. Cada lista contiene PSI ordenadas por nivel de impacto (de mayor a menor)

ID	Categoría
A	Accesos no autorizados
B	Malware
C	Filtración de datos
D	Denegación de servicios
E	Incumplimiento de políticas de seguridad

Valor del impacto	Nivel del Impacto
1 a 4	Muy bajo
5 a 10	Bajo
11 a 14	Medio
15 a 17	Alto
18 a 20	Muy Alto

Se muestra a continuación un ejemplo de cómo se recibirían los PSI:

Accesos no autorizados	A	A	A	A	
	15	13	12	11	
Malware	B	B			
	20	19			
Filtración de datos	C	C	C		
	18	17	16		
Denegación de servicios	D	D	D	D	D
	17	15	14	13	12
Incumplimiento de políticas de seguridad	E	E	E	E	
	14	12	10	8	

El EGPSI quiere reorganizar los PSI que atenderá en la semana, considerando los siguientes lineamientos:

1. Por cada categoría se genera una lista que tiene ordenados los PSI según su nivel de impacto. Los niveles de impacto no se repiten en una categoría.
2. Se generará una sola lista con todos PSI reorganizados por nivel de impacto y en orden de ID de categoría.
3. Si hay 2 PSI con el mismo nivel de impacto, se ordenan según la categoría. Por ejemplo: si 2 PSI de categorías A y D tienen nivel 15 de impacto, en la lista final iría primero el de categoría A y luego el de D.

Tomando en consideración el ejemplo anterior, la lista reorganizada de PSI, para que sea atendida en la semana por el EGPSI, sería la siguiente:

Lista	B	B	C	C	D	C	A	D	D	E	A	D	A	D	E	A	E	E
reorganizada	20	19	18	17	17	16	15	15	14	14	13	13	12	12	12	11	10	8

Se le solicita:

- a) Diseñar la estructura de datos que soporte la metodología de trabajo del EGPSI e implemente el ingreso de datos por teclado o archivo de las 5 listas que corresponden a las 5 categorías de problemas de seguridad de la información. (2 puntos).
- b) Implementar el algoritmo para reorganizar las 5 listas en 1 sola (no debe crear una lista nueva, emplee una de las listas originales) (7 puntos).

**Para la reorganización de las listas se debe tener en cuenta lo siguiente:**

- Si 2 categorías tienen PSI que no coinciden o interceptan en niveles de impacto (por ejemplo las listas de Malware y Filtración de datos) su fusión debería ser de complejidad  $O(1)$ . Para la fusión de 2 categorías en general la complejidad no debe superar  $O(n)$ .
  - No se puede utilizar memoria extra (es decir no puede usar new()) en ninguna parte, ya que el objetivo del problema es que mueva los nodos, no que los borre y/o cree nuevamente), tampoco puede ser estructuras adicionales, tipos abstractos de datos (TADs), arreglos o matrices.
- c) Implementar la función que permita mostrar la lista final resultado de la reorganización (1 punto).

Profesores del curso:

David Allasi  
Fernando Huamán  
Rony Cueva

San Miguel, 27 de mayo del 2023