

Asignatura: Estructura de Datos.

Responsable: Julio Cesar Martinez Ramírez.

Grupo: 302.

Instrucciones:

1. Lea detenidamente los enunciados y algoritmos solicitados.
2. Las dudas generadas se explicarán 5-10 minutos previos a comenzar el examen.
3. El examen comenzara después de solucionar las dudas del mismo.
4. La duración del examen es de 1hr y 50 min.
5. Las respuestas y/o algoritmos se enviarán al profesor como mensaje privado y/o e-mail una vez terminado el examen.
6. El formato del archivo enviado es: ED_<alumno>_1P.txt
7. Se sugiere usar SublimeText o algún otro editor de texto de su preferencia.
8. El documento de respuestas deberá incluir una línea de texto como comentario (usar //) por cada ejercicio, replicando el código de pregunta. Por ejemplo: // P1.
9. Considere todas las validaciones necesarias para los algoritmos.
10. **Para considerarse una respuesta válida, se debe responder en código o pseudocódigo con comentarios explicando el funcionamiento del algoritmo. En caso contrario no se tomará en cuenta.**

Colas:

- 1) (2.5pts) Dada la siguiente estructura:

```
struct Componente{
    AlgunElemento elemento;
    struct Componente* siguiente;
}
typedef Componente * Puntero;
struct Cola {
    Puntero principio, final;
}
```

Implementar las siguientes operaciones:

- Crear una cola: inicializarla
- Función PRIMERO: devuelve el elemento que ocupe la primera posición
- ENCOLAR: introduce un nuevo elemento a la cola

- **ELIMINAR:** eliminar el elemento de la cabeza de la cola
- **VACIA:** Averigua si la cola no tiene ningún elemento en su interior.

- 2) (2.5pts) **Contexto:** Se tiene una cola en la cual se han repartido tickets con el orden de atención. Sin embargo, llegada la hora de inicio hay muchos “colados”, es por esto que se le ordena al vigilante que retire a todos aquellos que no tienen ticket. Muestre la cola inicial, qué elementos fueron retirados de la cola y la cola final. Sugerencia: desencole cada elemento, si tiene ticket se vuelve a encolar, sino se retira. Note que los elementos se retiran y se reingresan según alguna condición.

Realice una función VIGILANTE que reciba una cola inicial y que regrese la cola sin considerar a los “colados”.

Pilas:

- 1) (2pt) **Contexto:** Un almacén tiene capacidad para apilar n contenedores. Cada contenedor tiene un número de identificación. Cuando se desea retirar un contenedor específico, deben retirarse primero los contenedores que están encima de él y colocarlos en otra pila, efectuar el retiro y regresarlos. Codifique los métodos Push () y Pop () para gestionar los contenedores.
- 2) (2pts) **Contexto:** Suponga que se dispone de un compilador que ya tiene implementado una implementación de estructura de datos: pilas y sus siguientes métodos:
- a. void INICIALIZAR(Pila* pila)
 - b. void APILAR(Pila* pila, Entero elemento)
 - c. void SACAR(Pila * pila)
 - d. Entero Cima(Pila* pila, Entero elemento)
 - e. BOOLEAN Vacia(Pila* pila)

Se debe implementar lo siguiente:

Escriba una función que invierta una pila utilizando las operaciones descritas anteriormente. NOTA: Se podrán utilizar todas las funciones auxiliares que tenga implementadas el compilador. Se invertirá el contenido de la pila utilizando una pila auxiliar.

- 3) (1pts) **Contexto:** Un conductor maneja de un pueblo origen a un pueblo destino, pasando por varios pueblos. Una vez en el pueblo destino, el conductor debe regresar a casa por el mismo camino. Utilice una estructura pila, un mínimo de 5 pueblos y muestre el camino recorrido tanto de ida como de vuelta.