

EJERCICIOS PARA EL PRIMER PARCIAL (de exámenes pasados)

1. Ordenar el siguiente arreglo de enteros utilizando los métodos de la burbuja y quicksort. Indique el número de intercambios realizados por cada algoritmo e indique cuál es más eficiente

5	3	7	6	4	2
---	---	---	---	---	---

2. Utilizando las operaciones primitivas del TDA Pila, implante la operación mezclar, que toma dos pilas ordenadas ascendentemente y genera una tercera pila que contiene los elementos de las dos pilas dadas también ordenados en forma ascendente.

Las operaciones primitivas de Pila que podrá utilizar son:

Tope: Pila → Entero
Retorna el elemento del tope de la pila
Apilar: Pila x Entero → Pila
Retorna la pila con el nuevo elemento en el tope
Desapilar: Pila → Pila
Retorna la pila sin el tope
EsVacía: Pila → Lógico
Retorna verdadero si la pila es vacía y falso en caso contrario
CrearVacía: → Pila
Crea una nueva pila vacía

3. Implemente un procedimiento que dada una lista enlazada simple de enteros, la ordene en forma ascendente. Considerar que pueden existir elementos repetidos.
4. Indique si las siguientes afirmaciones son ciertas o falsas. Justifique su respuesta :
- Si un problema puede resolverse mediante procedimientos recursivos y no recursivos, el recursivo es siempre más eficiente.
 - La especificación de las primitivas definidas para un TDA cambia según la estructura (implementación) utilizada.
 - La implementación de listas mediante apuntadores es siempre más eficiente que con arreglos (considerar espacio de memoria y tiempo de acceso).
5. Haga una corrida del siguiente programa, indicando las llamadas realizadas, el valor de las variables locales en cada ambiente de referenciación, y el valor resultante.

Función $CB(N, M: \text{entero}) \rightarrow \text{entero}$

Si $N=M$ o $M=0$ entonces

Retorna 1

Sino

Retorna $CB(N-1, M-1) + CB(N-1, M)$

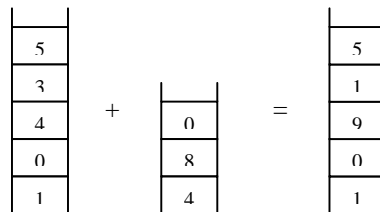
Fin

Fin

Llamada original: $CB(4,2)$

6. Dados dos enteros positivos A y B colocados dígito a dígito en sendas pilas de forma tal que el dígito menos significativo se encuentre en el tope de la correspondiente pila, se desea que usted diseñe un algoritmo recursivo que retorne otra pila con la suma de ambos enteros. Ej:

$$10435 + 480 = 10915$$



Las operaciones primitivas de Pila que podrá utilizar son:

CrearPila: \rightarrow Pila

Crea una pila vacía.

EsVacía: Pila \rightarrow Lógico

Función que retorna verdadero si la pila es vacía y falso en caso contrario.

Tope: Pila \rightarrow Elem

Función que retorna el elemento del tope de la pila.

Desapilar: Pila \rightarrow Pila

Función que retorna una pila que contiene los mismos elementos de la pila original menos el tope.

Apilar: Elem x Pila \rightarrow Pila

Función que retorna la pila con el elemento dado en el tope de la misma.

7. Una compañía que se encarga de hacer el transporte de niños de ida y vuelta entre su casa y su colegio requiere un programa que se encargue de mantener actualizadas las listas de niños que usan cada una de las 10 rutas que mantiene la compañía. Cada ruta está identificada por un número consecutivo del 1 al 10 y no existe límite predefinido de niños asignados a cada ruta (se asignarán a cada ruta tantos autobuses como sean necesarios). Por cada niño se necesita conocer su nombre, apellido, grado, dirección y el código numérico del colegio al que va. En base a esta información se requiere que usted:
- Plantee una estructura de datos eficiente, que permita almacenar la información requerida.
 - Implemente las siguientes operaciones de acuerdo a la estructura especificada:
 - Inscribir a un niño en una ruta determinada.
 - Retirar a un niño de una ruta determinada.
 - Consultar la cantidad de niños inscritos en cada una de las rutas.

8. Escriba un programa que lea una línea de la entrada y determine si la frase leída es un palíndromo. Una frase se dice que es palíndromo si se puede leer lo mismo de izquierda a derecha y de derecha a izquierda, una vez eliminados los blancos. Un ejemplo es "dábale arroz a la zorra el abad" (Recomendación: Utilice una pila y una cola)

9. Un método de ordenación se dice que es estable si dos elementos que tienen la misma clave permanecen en el orden en el que aparecieron en la entrada. Indique qué métodos de ordenación dados (selección, quicksort, burbuja) son estables. Use la siguiente entrada para determinarlo:

12	34	56	43	22	34	18	96	55	12	28	10	48
----	----	----	----	----	----	----	----	----	----	----	----	----

10. Ejecute el algoritmo Quicksort en el siguiente arreglo de elementos. Precise claramente el árbol de ejecución detallando las llamadas recursivas y los valores de las variables.

60	11	36	24	35	32	32	25
----	----	----	----	----	----	----	----

11. En un salón de juegos de Ajedrez para niños se ha diseñado un sistema para determinar diariamente, el orden de selección de niños en grupos de a dos. Para ello los coordinadores han decidido seguir la siguiente estrategia: cada día se formará un círculo con todos los niños en orden de llegada de éstos (numerados del 1 al N, siguiendo el sentido de las agujas del reloj).

Comenzando desde 1 y moviéndose en sentido horario, uno de los coordinadores cuenta "k" niños, mientras el otro coordinador comienza en N y se mueve en sentido antihorario contando "m" niños. Estos dos niños son elegidos. Si los dos coordinadores coinciden en el mismo niño se toma el niño siguiente en el sentido horario. Cada uno de los coordinadores comienza a contar de nuevo en el siguiente niño que permanezca en el círculo y el proceso continua hasta que no quede nadie. El grupo de niños siempre es un número par y no excede de 30 niños. Se pide:

- Diseñe las estructuras apropiadas, tomando en cuenta que el resultado final debe reflejar las "parejas" de niños en el orden en que fueron seleccionados.
- Diseñe un algoritmo para la selección de las parejas.

12. Diseñar un algoritmo utilizando el método de intercambio que dada una lista circular que contiene números enteros los ordene en forma descendente. El ordenamiento debe hacerse sin utilizar ninguna estructura auxiliar

13. En una pila se tienen los coeficientes de un polinomio de grado N

1.2	(Tope)
-3.6	
0	
2.6	
12	

Esta pila representa a los coeficientes del polinomio :

$$P(x) = 12x^4 + 2.6x^3 - 3.6x + 1.2$$

Cree un función que evalúa, utilizando únicamente operaciones de pila, al polinomio en un punto x cualquiera utilizando la **Regla de Horner**:

$$P(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + xa_n) \dots))$$

Operaciones válidas:

Tope(Pila A), Desapilar (Pila A), Vacío (Pila A), Apilar(Pila A, Número N)

Restricciones: La pila debe quedar intacta al final y no se puede utilizar una pila auxiliar.

14. Dada una lista enlazada de enteros. Realice el procedimiento recursivo que determina la longitud de la secuencia creciente más larga que contiene. Por ejemplo:

La lista: 3 5 11 2 9 15 10 9 9 19 22 24 18

En este caso la respuesta es 5 (secuencia: 9 9 19 22 24)

15. Un dipolo es una estructura de lista que tiene como indicador una variable estructurada que contiene dos punteros simples a casillas de lista, uno apunta a la cabeza de la lista y el otro a la cola de la misma :

```
Struct dipolo {
    Nodo *cabeza;
    Nodo *cola }
```

Implemente las siguientes primitivas del **TDA lista** utilizando dipolos.

- **Insertar (Dipolo A, Entero N)** (Inserta ordenadamente el entero N en la lista A)
- **Eliminar (Dipolo A, Entero N)** (Elimina el entero N, de la lista A)

16. Muestre el árbol de ejecución del algoritmo siguiente:

Función RC(r: entero, x: entero) → entero

si (r = -1) entonces

retornar -1;

sino

si (M[r].A = x) entonces

retornar r;

sino

si (M[r].A > x) entonces

retornar RC(M[r].B, x);

sino

retornar RC(M[r].C, x);

fsi

fsi

fsi

ffunción

Tipo T = registro

A : entero;

B : entero;

C : entero;

fregistro

Var M: arreglo [0..15] de T;

Los valores de la variable M son los siguientes:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	50	25	10	5	7	45	30	28	35	70	80	75	71	77	90	85
B	1	2	3	-1	-1	6	7	-1	-1	-1	11	12	-1	-1	-1	-1
C	9	5	-1	4	-1	-1	8	-1	-1	10	15	13	-1	-1	-1	14

La llamada original es: $r \leftarrow RC(0, 35)$

17. En una cárcel de alta seguridad, se encuentran los prisioneros más peligrosos que han sido condenados a muerte. Existe en la penitenciaría, un sistema automatizado que lleva registrados los condenados a muerte en una "lista", donde el primer elemento es el preso más antiguo y éste tiene el número uno (1). Cuando se activa la pena de muerte en el país, se activa un proceso de "selección" que escoge de la lista los presos que se ejecutarán, Como la ley humana no es infalible quieren dejar algo para la ley de Dios, Así que este proceso se conoce como "SUPERVIVIENTES" ya que comienza seleccionando un preso de cada "n" (, por ejemplo, si $n = 2$ el preso 1, 3, 5,.....), luego, con la lista resultante escoge uno de cada "n+1", luego, uno de cada "n+2", y así sucesivamente. El proceso termina cuando se va a eliminar un número de cada "m" y el tamaño de la lista es menor a "m". Los presos que se queden en la cola en ese momento son los supervivientes. Se pide diseñar "SUPERVIVIENTES" recursivo que reciba la lista de los presos y un n inicial. El proceso debe devolver a los sobrevivientes en una cola de manera que el frente de la misma contenga el primer afortunado.

Las operaciones primitivas de que podrá utilizar para las colas son: **Frente**, **Encolar**, **Desencolar**, **Colavacia** y **CrearCola**:

18. Una expresión aritmética construida con los operadores aritméticos binarios '+', '-', '*', '/' y operandos de un único dígito entre 0 y 9, se dice que está en forma pos-fija cuando el operador binario está al final de los dos operandos. Los operandos a su vez que lo conforman pueden ser a su vez, expresiones en forma post-fija, aplicando el mismo concepto. Se asumirá que la expresión está totalmente agrupada entre paréntesis para no tener problema con la escogencia de prioridad de los operadores. A continuación se muestra un ejemplo:

La expresión aritmética:

$((((6 + 4) / 5) + (8 * (3 / (5 * 2))))))$ (forma in-fija)

Tiene la forma post-fija: $6\ 4\ +\ 5\ /\ 8\ 3\ 5\ 2\ *\ /\ +$

Se asume que la expresión está en una pila donde cada nodo contiene un elemento que puede ser un paréntesis, un número o un operador aritmético. En el tope de la pila está el elemento más a la izquierda de la expresión (en el ejemplo, el tope sería "("). Se asumirá también que la expresión está bien construida.

Se pide:

Diseñe un algoritmo que dada la pila que contiene la expresión en forma in-fija, construya la expresión en forma post-fija y la deje en otra pila (donde el tope será también el elemento más a la izquierda).

Puede usar estructuras auxiliares (dinámicas)

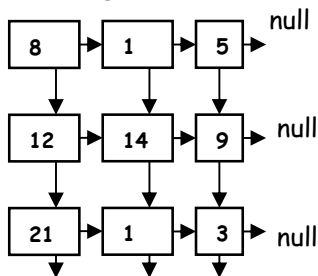
18. Una cola se representa con la siguiente estructura de datos

```
struct tcola {
    int final, frente;
    int elementos [50]
}
```

Si no se guarda el número de elementos de la cola, si frente = 38 y final = 36, ¿Cómo está la cola?

- i) La cola está completamente llena
- ii) La cola está vacía.
- iii) La cola está casi llena excepto un elemento.
- iv) La cola está casi vacía
- v) Ninguna de las anteriores

19. Una matriz puede representarse como una estructura dinámica enlazando filas y columnas de la siguiente forma:



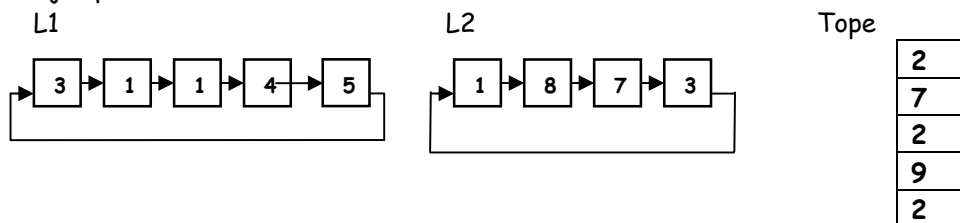
Se define el punto de ensilladura como la posición i,j que contiene al mínimo valor de la fila i que corresponde al máximo valor de la columna j.

En el ejemplo, la matriz posee un punto de ensilladura en la posición 2,3 (mínimo valor de la fila 2 que corresponde al máximo valor de la columna 3). Escriba una función que escriba los valores de i y de j del punto de ensilladura de una matriz dada en el caso de existir. En caso contrario debe devolver los valores -1,-1.

20. Dos números enteros se encuentran representados en dos listas circulares simplemente enlazadas, de forma tal que cada nodo contiene un dígito entre 0 y 9. El dígito más significativo se encuentra en la primera posición de la lista.

Se pide implementar una función RESTA, que reciba como parámetro las dos listas y devuelva como resultado una pila con la resta de ambos números. En el tope de la pila quedará el dígito menos significativo.

Ejemplo: 31145 - 1873 = 29272



No se permite utilizar estructuras auxiliares y la pila debe trabajarse solo con las primitivas vistas en clase.

21. Dada una pila que contiene enteros, se pide desarrollar un procedimiento recursivo que elimine todos los elementos duplicados. Ejemplo: la pila con los elementos: 1, 5, 3, 2, 2, 5, 7, 2 quedaría tras la aplicación de la operación como: 1, 5, 3, 2, 7.

Tope	1	
	5	
	3	
	2	1
	2	5
	5	3
	7	2
	2	7

Use solamente las primitivas vistas en clase. No puede utilizar estructuras auxiliares

22. Se aplicó la función Hashing: $H(\text{clave}) = (\sum d_i) \text{ MOD } 11$ y se utilizó con estrategia de manejo de colisiones el método la prueba lineal o secuencial. Sobre la tabla de 11 posiciones se insertaron siete claves, como se muestra a continuación:

787				103	933	899	475	676	045	
0	1	2	3	4	5	6	7	8	9	10

Indique cuál de las siguientes afirmaciones es falsa:

- La clave 899 colisionó al momento de la inserción
- Un posible orden en que las claves se insertaron es: 103, 475, 787, 045, 933, 676, 899
- La inserción de la clave 676 no provocó colisión alguna
- La búsqueda de la clave 103 requiere sólo una comparación
- Ninguno de los anteriores.