

ELEMENTOS DE PROGRAMACIÓN

Trabajo Práctico N° 6: Uso de TDA Lista , TDA Pila y TDA Cola.

1. Desarrolle un módulo de prueba y defina un conjunto de datos para testear todas las operaciones de las librerías TDA Pila, TDA Cola y TDA Lista provistas en la práctica.
2. Escriba un subprograma para alterar el contenido de una lista L reemplazando todos los elementos ubicados en posiciones pares por un valor X.
3. Represente un número binario a través de una lista y escriba un procedimiento INCREMENTAR (num_bin) que sume 1 al número binario num_bin.
4. Escriba un subprograma, utilizando el TDA Pila, para evaluar la correctitud de una expresión matemática, en cuanto a la utilización de { [(.
5. Use una pila para eliminar la recursión del siguiente procedimiento:

```
PROCEDURE Invertir (VAR L:Lista);
VAR x : TipoElemento;
BEGIN
    IF NOT TamanoLista(L) = 0 THEN
        BEGIN
            Obtener(L, 1, x);
            Suprimir(L, 1);
            Invertir(L);
            Insertar(L, TamanoLista(L), x);
        END
    END
END
```

6. En una cola de pedidos se han alojado solicitudes de dos tipos distintos (A y B). Escribir un procedimiento para separar los pedidos tipo A de los pedidos tipo B en dos colas distintas, respetando el orden que tenían en la cola original:
 - Usar la cola que ya estaba creada para alojar las solicitudes tipo A y una cola nueva para las solicitudes tipo B.
 - No se conoce previamente el número de solicitudes que hay en la cola original de pedidos.
 - Al finalizar la ejecución del programa deberá aparecer en pantalla el número de solicitudes de cada tipo.
7. Dado un texto codificado, almacenado en un string, se pide desarrollar un algoritmo que pase carácter por carácter del string a una lista y genere otro string con el texto decodificado.

La codificación consiste en: cada palabra se encuentra entre pares "ZX" y "XZ" y está en forma invertida.

Ejemplo: el string original contiene:

PZZXATSEXZYTZXERBOSXZ2Z6ZXLEXZUYZXOIROTIRCSEXZ2PP8U.

El string resultado será: ESTA SOBRE EL ESCRITORIO.

8. Desarrollar un procedimiento que reciba como parámetros de entrada una cola **Q1** de caracteres con el siguiente formato :

Q1 = \leftarrow Subcad-1#Subcad-2#...#Subcad-n \leftarrow

donde cada Subcad-i es una sucesión de letras mayúsculas de la 'A' a la 'Z'.

Y devuelva otra cola **Q2** con el siguiente formato:

Subcad-1#(Subcad-1)'#Subcad-2#(Subcad-2)'#...#Subcad-n#(Subcad-n)'

donde cada (Subcadena-i)' representa la Subcadena-i invertida.

Ejemplo: Si la Entrada es : **Q1** = \leftarrow A B # C # D E F # G \leftarrow

Entonces la salida deberá ser así:

Q2 = \leftarrow A B # B A # C # C # D E F # F E D # G # G \leftarrow

9. Desarrollar un procedimiento que reciba como parámetros de entrada una cola **Q1** de caracteres y un número natural **n**, y devuelva otra cola **Q2** donde el orden de los bloques de **n** caracteres de **Q1** está invertido en **Q2**. Además, cada bloque en **Q2** deberá conservar su orden original . Considere como precondition que la longitud de **Q1** es múltiplo de **n**.

Ejemplo: Entrada: **Q1** = \leftarrow T A C O \leftarrow y **n** = 2

Entonces, hay dos bloques de 2 letras: TA y CO, En la cola de salida **Q2** deberán quedar invertidos así: **Q2** = \leftarrow C O T A \leftarrow

10. Escriba un subprograma para determinar si la cadena de caracteres que se ingresa es de la forma: **a₁ XX a₁' XX a₂ XX a₂' XX ... a_n XX a_n'**, con $n > 0$, donde **a_i'** es la cadena **a_i** invertida y, a su vez, **a_i** es una cadena de la forma **bZbb**, donde **b** pertenece al conjunto **{A,B}*** (que representa el conjunto de todas las cadenas constituidas únicamente por los símbolos pertenecientes al conjunto **{A,B}**)

Ejemplos de cadenas válidas:

BZBBXXBBZB

ABBZABBABBXXBBABBABZBBA

AABAZAABAAABAXXABAAABAAZABAAXX BZBBXXBBZB

Operaciones del TDA Lista

PROCEDURE CrearLista(VAR L: Lista);
FUNCTION TamanoLista(L: Lista): Word;
PROCEDURE Obtener(L: Lista; Posi: Integer; VAR X: TipoElemento);
PROCEDURE Suprimir(VAR L: Lista; Posi: Integer);
PROCEDURE Insertar(VAR L: Lista; Posi: Integer; X: TipoElemento);
FUNCTION Localizar(L: Lista; X: TipoElemento): Integer;
PROCEDURE InsertarCreciente(VAR L: Lista; X: TipoElemento);
PROCEDURE InsertarDecreciente(VAR L: Lista; X: TipoElemento);

Operaciones del TDA Pila

PROCEDURE CrearPila(VAR P: Pila);
FUNCTION PilaVacía(P: Pila): Boolean;
PROCEDURE Tope(P: Pila; VAR X: TipoElemento);
PROCEDURE Sacar(VAR P: Pila; VAR X: TipoElemento);
PROCEDURE Poner(VAR P: Pila; X: TipoElemento);

Operaciones del TDA Cola

PROCEDURE CrearCola(VAR Q: Cola);
FUNCTION ColaVacía(Q: Cola): Boolean;
PROCEDURE Qfrente(Q: Cola; VAR X: TipoElemento);
PROCEDURE Qsacar(VAR Q: Cola; VAR X: TipoElemento);
PROCEDURE Qponer(VAR Q: Cola; X: TipoElemento);