ALGORITMOS Y ESTRUCTURAS DE DATOS

Trabajo Práctico no. 5 **Fecha: 25/04/24**

Tema: **Tipo de datos PILA**

1. Considere el ADT PILA(item) visto en clase:

   a. Agregue a la ***especificación algebraica*** del mismo las siguientes operaciones:

      i. FONDO: selectora que devuelve el item que se encuentra en el fondo de la pila.
      ii. POPF: modificadora que elimina el ítem que se encuentra en el fondo de la pila.
      iii. REEMP: modificadora que reemplaza todas las ocurrencias de un dado ítem por otro.

   b. Como ***usuario*** del ADT diseñe una función recursiva **eliminarNegativos** que dada una pila de números enteros elimine de la pila todos los elementos negativos y retorne la pila resultante.

   c. Codifique en un archivo de nombre **Pila.h** una implementación en lenguaje C del ADT PILA de enteros con lista enlazada utilizando la tipificación vista en la clase práctica. Escriba un programa para probar todas las operaciones del ADT Pila y la operación del apartado b). Calcule la complejidad de las operaciones en notación O Grande

2. Teniendo en cuenta las operaciones del ADT PILA(item): PILAVACIA (PV), PUSH, POP, POPF, TOP, PUSHF, FONDO y REEMPLAZAR (Reemp) indique en función de las constructoras primitivas cuál es la pila resultante en las siguientes expresiones:

a. PUSHF(PUSH(PV,c), FONDO(PUSH(POPF(PUSH(PUSH(PV,e), f)),g)))

b. REEMP(PUSH(POP(PUSHF(PUSH(PUSH(PV, a), b), c)),a) , a, TOP(PUSHF(PUSH(PV, J), k)))

3. **Como usuario del ADT PILA** escriba el algoritmo para **CONVERTIR** una expresión aritmética dada en notación infija a una expresión en notación posfija*****. El proceso de convertir acepta una expresión infija como entrada y produce una expresión posfija como salida. Considere expresiones bien formadas que tengan variables (de la 'a' a la 'z'), operadores binarios (+, -, *, /), el operador unario (~) y paréntesis terminadas por la marca final '='. La idea es utilizar una pila para almacenar los operadores a medida que son encontrados para más tarde desapilar estos operadores de acuerdo a su precedencia.

Considere los siguientes ***casos de prueba***:

forma infija	forma posfija
a+b	a b +
~a+b	a ~ b+
a+b*a	a b a * +
(a+ (~b))*c	a b ~ + c *
a*(b+c)+a/e	a b c + * a e / +
a+b-c	a b + c -

forma infija	forma posfija
(a+(~b-c))	a b ~ c - +
(a+b)-c	a b + c -
(a-(b+c))	a b c + -
(b-a)/(c+d)	b a - c d + /
a+b/(d-a)*e	a b d a - / e * +
a*b/c	ab*c/