Guion de la Practica 5 de Algoritmos y Estructura de Datos

Objetivo

El objetivo de esta práctica es poner en práctica las competencias adquiridas durante el desarrollo de los temas relacionados con los Tipos Abstractos de Datos (TAD). En concreto se hará hincapié en los conceptos de **listas**, **pilas** y **colas**, al implementar una calculadora para evaluar expresiones RPN (*Reverse Polish Notation*)

Material

Tendrán a su disposición el siguiente material:

- `dll_node_t.hpp`: clase nodo de la `dll_t`.
- `dll_t.hpp`: clase lista doblemente enlazada.
- `stack_l_t.hpp`: clase pila implementada con `dll_t`.
- `queue_l_t.hpp`: clase cola implementada con `dll_t`.
- `rpn_t.hpp`: clase que evalúa la RPN.
- `main_rpn_t.cpp`: fichero principal para probar la práctica.
- `data_rpn_t_1.txt`, `data_rpn_t_2.txt`, `data_rpn_t_3.txt`: datos de prueba.

FASE I: COMPRENSIÓN DEL MATERIAL PROPORCIONADO Y DESARROLLO DE LOS OPERADORES BÁSICOS

Deben leer y comprender el material proporcionado, y estudiar en profundidad la sintaxis de cada uno de los los ficheros. Es muy importante que se comprenda las dos clases TAD proporcionadas: pila y cola basada en lista doblemente enlazada. Debe estudiarse el mecanismo de la notación polaca inversa, así como la implementación de una calculadora utilizando este tipo de notación mediante una pila.

En este punto se debe desarrollar la calculadora RPN para los operadores de aridad dos **`+`**, **`-`**, **`*`** y **`/`**. Para ello, este enlace muestra un ejemplo con traza de cómo funciona el algoritmo que se pretende desarrollar.

FASE II: DESARROLLO DE OPERADORES MÁS AVANZADOS

En este punto debe desarrollarse el operador de aridad dos **`^`** (**exponente**) y el operador de aridad uno **raíz cuadrada** (que denotaremos con el símbolo **`r`**),

FASE III: DESARROLLO DE OPERADORES MÁS AVANZADOS DE ARIDAD UNO

Se debe completar la implementación de la calculadora RPN para los operadores de aridad uno **logaritmo en base 2** (`l`) y **elevación al cuadrado** (`c`).

Evaluación

La calificación de la práctica dependerá de la calidad y eficiencia de la misma.

- Si la práctica funciona correctamente, así como la modificación propuesta por el profesor, y ha concluido la **Fase I**: **hasta 5**
- Si la práctica funciona correctamente, así como la modificación propuesta por el profesor, y ha concluido la **Fase II**: **hasta 7**
- Si además de lo anterior, se ha efectuado la **Fase III**: **hasta 10**

Una vez presentada la modificación de la práctica (y no antes) **deben ser enviados solo los ficheros cpp y hpp** a través del enlace de esta tarea al campus virtual.

Resultado esperado

Si ejecutan la práctica con todas las fases desarrolladas, la salida esperado por pantalla sería:

```
$ main_rpn_t < data_rpn_t_3.txt
Expresión a evaluar: - 3 + * 4 + 2 1 5
Sacamos de la cola un carácter: 5 (es un dígito)
Lo metemos en la pila...
Sacamos de la cola un carácter: 1 (es un dígito)
Lo metemos en la pila...
Sacamos de la cola un carácter: 2 (es un dígito)
Lo metemos en la pila...
Sacamos de la cola un carácter: + (es un operador)
Sacamos de la pila un operando: 2
Sacamos de la pila otro operando: 1
Metemos en la pila el resultado: 3
Sacamos de la cola un carácter: 4 (es un dígito)
Lo metemos en la pila...
Sacamos de la cola un carácter: * (es un operador)
Sacamos de la pila un operando: 4
Sacamos de la pila otro operando: 3
Metemos en la pila el resultado: 12
Sacamos de la cola un carácter: + (es un operador)
Sacamos de la pila un operando: 12
Sacamos de la pila otro operando: 5
Metemos en la pila el resultado: 17
Sacamos de la cola un carácter: 3 (es un dígito)
Lo metemos en la pila...
Sacamos de la cola un carácter: - (es un operador)
Sacamos de la pila un operando: 3
Sacamos de la pila otro operando: 17
Metemos en la pila el resultado: 14
Resultado: 14
```