Examen 3:

“Evaluador de expresiones”

Documentación

Elaborado por:

Jean Paul Rodríguez Flores

Daniel Sequeira Retana

Estructuras de Datos

Grupo 02

Profesor Carlos Benavides Céspedes

7 de junio del 2020

Introducción

El objetivo de este examen es crear un árbol binario de expresiones a partir de una expresión matemática ya sea en notación postfija, prefija o infija, construyendo un árbol para la solución y además proporcionando el resultado de dicha operación. Además, debe contar con una interfaz que permita al usuario usar el programa de manera más sencilla, las operaciones matemáticas a usar serán las comunes de una calculadora simple además de las opciones avanzadas que la calculadora científica provee, como la raíz enésima, elevar, logaritmo, etc.

Descripción de la solución

El uso de árboles binario fue fundamental para resolver el problema, teniendo así un manejo ordena también proporcionando una manera rápida de acceder a todos los órdenes de recorridos. Para la resolución de examen se usaron las siguientes TDA:

-TDA NodoLista

Una pequeña estructura que sirve como nodos para listas simples utilizadas en el procesamiento de los árboles. No posee funciones y guarda un QString, y un puntero de su mismo tipo para hacer las listas doblemente enlazadas

-TDA ListaSimple

Lista simple doblemente enlazada básica, posee las siguientes funciones

a) void add(Qstring dato): añade un nodo a la lista con el dato que se pasó por parámetro
b) QString imprimir(): imprime toda la lista simple y además retorna toda la lista como un solo QString

-TDA NodoArbol

Un nodo de árbol común con su parte derecha, izquierda y el dato de tipo QString que guarda. Posee las siguientes funciones:

a) Void visitar(NodoArbol * nodo): imprime en consola el QString almacenado del nodo árbol
b) Void inorden(NodoArbol *raíz): imprime el árbol en recorrido inorden
c) Void postorden(NodoArbol *raíz): imprime el árbol en recorrido postorden
d) Void preorden(NodoArbol *raíz): imprime el árbol en recorrido preorden
e) ListaSimple *listaEnorden(QString num): crea una lista simple a de cada carácter de un QString
f) ListaSimple *listapreorden(NodoArbol *raiz,ListaSimple *qLista): crea una lista simple de un árbol en recorrido preorden
g) ListaSimple *listapostorden(NodoArbol *raiz,ListaSimple *qLista): crea una lista simple de un árbol en recorrido postorden
h) ListaSimple *listainorden(NodoArbol *raiz,ListaSimple *qLista): crea una lista simple de un árbol en recorrido inorden

-TDA NodoPila

Un nodo para una pila que almacena nodo árboles. No posee funciones propias

-TDA Pila

Un pila que almacena nodos de tipo NodoPila, esto es para evitar tener que poner un puntero a siguiente en los Nodos árboles. Posee las siguientes funciones

a) Void push(NodoArbol * nodo): mete a la pila un nodo Pila con el dato del parámetro
b) Nodo * pila pop(): elimina y retorna el nodo tope de la pila
c) Nodo * pila getTope(): retorna el nodo tope de la pila
d) Bool empty: retorna un true si la pila está vacía, un false en case contrario
e) Void imprimir(): imprime toda la pila

-TDA PilaSimple

Un pila que almacena nodos de tipo NodoLista, para manejar pilas con otros tipos de datos, posee las mismas funciones que Struct Pila

Análisis de resultados

El examen 3 fue completado con éxito en todos sus aspectos, utilizando las funciones matemáticas requeridas, creando arboles y pasando de expresiones a otras sin ningún problema, además que se crearon funciones desde 0, sin ayuda externa para crear algunos algoritmos de conversión a otros tipos de expresiones

Conclusiones

Las expresiones prefijas y postfijas a pesar de no ser de uso cotidiano presentan diferentes ventajas que las infijas no pueden darse, y aunque no lo parezca su dificultad no va más allá de una expresión en inorden, por lo que es bastante conveniente conocer algoritmos para convertir las expresiones en todos sus órdenes, con respecto al proyecto se cumplió el objetivo de este al reforzar los conocimientos en C++ y el manejo de árboles.

Bibliografía

Algoritmos y estructuras de datos en C, página 431, Reglas para la construcción de un árbol de expresión