

Resumen

El proyecto se realiza con el fin de desarrollar una aplicación de consola en el lenguaje C++ que funcione como simulador de una Calculadora que trabaje con el algoritmo de pilas, utilice la Notación Polaca y la Polaca Inversa así también que logre convertir Valores Infijos a las otras dos Notaciones y las Resuelva.

Arreglos de Pilas Y Colas:

Son un Conjunto de arreglos los cuales a partir de ciertos punteros se van guardando de cierta manera la cual Puede ser en Forma de Pila O Cola.

- En el caso de la Pila que es el Arreglo que nos interesa este lo que hace es guardar la información debajo o detrás del último dato del arreglo sucesivamente
- Es importante decir que estos se recorren a partir de Punteros para funcionar y ser Modificados.

Expresión aritmética:

Formada por operando y operadores: $A*B / (A+C)$

- Operando: Variables que toman valores enteros o reales.
- Operadores: En caso de igualdad de prioridad. Son evaluados de izquierda a derecha (se evalúa primero el que primero aparece). $5*4/2 = (5*4) / 2 = 10$

Cuando aparecen varios operadores de potenciación juntos la expresión se evalúa de derecha a izquierda. $2^3^2 = 2^ (3^2) = 2^9 = 512$

Notación Prefija, Infija y Postfija

Notación Infija

Es la notación ya vista que sitúa el operador entre sus operandos.

- Ventaja: Es la forma natural de escribir expresiones aritméticas
- Inconveniente: Muchas veces necesita de paréntesis para indicar el orden de evaluación: $A*B/(A+C) \neq A*B/A+C$

Notación Prefija o polaca

En 1920 un matemático de origen polaco, Jan Lukasiewicz, desarrollo un sistema para especificar expresiones matemáticas sin paréntesis.

Esta notación se conoce como notación prefija o polaca (en honor a la nacionalidad de Lukasiewicz) y consiste en situar al operador ANTES que los operandos.

Ejemplo: la expresión infija $A*B / (A+C)$ se representaría en notación prefija como: $/*AB+AC$

Notación Postfija o Polaca Inversa

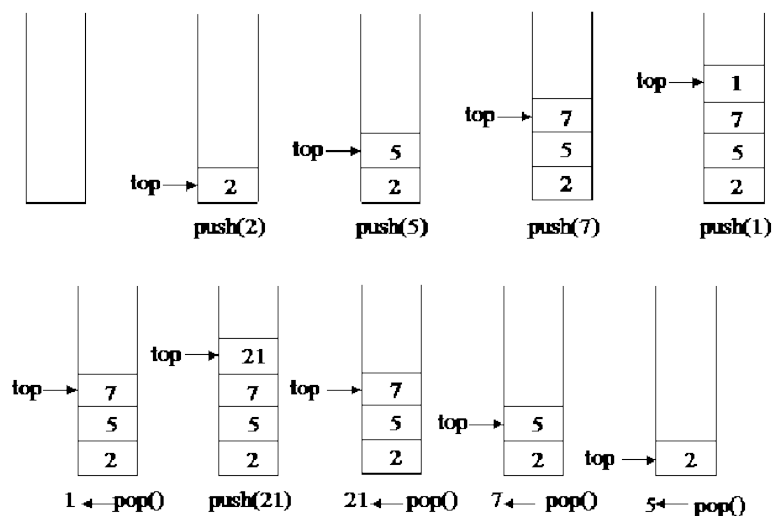
La notación postfija o polaca inversa es una variación de la notación prefija de forma que el operador se pone DESPUÉS de los operandos.

Ejemplo: la expresión infija $A*B / (A+C)$ se representaría en notación postfija como: $AB*AC+ /$

Ventajas:

- La notación postfija (como la prefija) no necesita paréntesis.
- La notación postfija es más utilizada por los computadores ya que permite una forma muy sencilla y eficiente de evaluar expresiones aritméticas (con pilas).

Dibujo que Ejemplifica el funcionamiento del Algoritmo de Pila Y la Notación Postfija trabajando en conjunto para resolver un problema Aritmético.



Manual del Software 1B:

Dentro del Archivo donde venía este documento viene un .cpp este se debe abrir copiar el contenido y pegarse en un nuevo proyecto en Visual Studio para ser ejecutado. Al ser ejecutado el programa nos muestra lo siguiente:

MENU DE CALCULADORA POSFIJA

1. Convertir de INFIJA a POSFIJA
2. Resolver una expresión en POSFIJA
3. Salir de la calculadora

En este Menú podemos escoger 3 Opciones Las cuales nos permite Convertir Expresiones infijas a Posfijas y Resolverlas, Resolver Una Expresión posfija o Salirnos del programa.

Si escogemos **La primer Opción** nos mostrara lo siguiente:

1. Ingresar una nueva expresión
2. Mostrar la última expresión ingresada
3. Volver al menú de calculadora

Acá nos permite ingresar una de 3 nuevas opciones, Expresión a Convertir y resolver, mostrarnos la última Ingresada y Volver al Menú.

Si escogemos **La Segunda Opción** del Menú Principal nos mostrara lo siguiente:

1. Resolver una expresión ingresada anteriormente
2. Ingresar una expresión nueva en INFIJO
3. Ingresar una expresión nueva en POSFIJO
4. Salir al menú de calculadora

En esta parte podremos Resolver la última expresión que ingresamos o resolver ya sea una Expresión Infija como Posfija.

En el caso de Tener una idea Vaga de que probar en la calculadora, se puede utilizar las expresiones agregadas al principio de este documento como teoría para probar cada una de las opciones que permite el software.

Funcionamiento del Código:

Este apartado explica el funcionamiento del código del Software.

- Primero se define un número máximo de caracteres.
- En el código se genera una estructura el cual se define el Operador en el cual se le define la prioridad el valor, carácter y un valor booleano de existencia. Esta Tendrá un Puntero.
- Con esto ya tenemos nuestro Arreglo
- Luego Tendremos una Función Para eliminar las listas(Pilas), esta funciona con un puntero auxiliar dentro de un ciclo.
- Luego se crea un operador a través de un puntero con 2 argumentos el tipo de carácter y la prioridad, este solicita memoria.
- Luego tenemos una Función que apila o agrega al final de la lista esta funciona con un puntero que recorre el arreglo hasta el último para insertar al final.
- Luego está la función que muestra en pantalla los arreglos a partir de un ciclo.
- Luego se utiliza una función la cual agrega al inicio de la pila
- Tenemos otra función la cual me recorre el arreglo y me indica el primero en la Pila
- Contamos con otra función que me extrae el primer valor y lo saca de la Pila; este funciona también con un puntero auxiliar con Condicionales.
- Se tiene otra función que me indica la prioridad de los operadores
- Se tiene una función que toma un string lo transforma en una lista enlazada del tipo operador
- A partir de dos punteros se crea una segunda lista que funciona como un conversor de infija a posfija esto funciona a partir de las prioridades definidas en una de las funciones previamente realizadas y varios condicionales

- Con un Puntero se crea otra función a partir de otros dos punteros auxiliares. Este a Partir de un ciclo Prueba condicionales a partir de las prioridades
- Al final de esta última función debe de retornar uno a uno el primero de la pila.