

Cambio a Notación Postfija

1. La notación habitual para expresiones algebraicas es conocida como Notación **Infija**, su principal inconveniente es que requiere del uso de paréntesis para romper la precedencia de las operaciones.
2. Una Expresión que emplea la notación **Prefija** nos indica que el operador va antes de los operandos, por ejemplo: + A B
3. Se evalúa de izquierda a derecha hasta que encontremos al primer operador seguido inmediatamente de un par de operandos, entonces se evalúa la expresión binaria y el resultado se cambia como un nuevo operando. El proceso se repite este hasta que nos quede un solo resultado.
4. Entonces la expresión prefija: * + A B C equivale a evaluar la expresión infija: (A+B) * C.
5. La notación **Postfija** es mucho más utilizada que la anterior y se refiere al caso en que el operador se coloca después de los operandos, por ejemplo: A B +
6. La operación postfija también se evalúa de izquierda a derecha, entonces la expresión: A B + C * equivale a la expresión infija: (A+B) * C
7. Antes de iniciar la práctica, localice en su carpeta de trabajo el archivo correspondiente al programa fuente llamado **ejercicio 15B.cpp** y ábralo con el editor de texto **WordPad**.
8. Borre las declaraciones **#include**, pero deje la línea siguiente:

```
using namespace std;
```
9. Borre la clase **Nodo** y su método constructor porque ya están definidos en **pila.h**, solamente deje intacta la clase **Cola** con todos sus métodos y borre el programa principal (main).
10. Finalmente guárdelo con el nombre de **cola.h** en su carpeta de trabajo.
11. Ejecute el programa **Dev-C++**.
12. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**, copie el siguiente encabezado, en el que incluimos las clases y métodos guardados en **pila.h** y en **cola.h**:

```
#include<iostream>
#include <iomanip>
#include "pila.h"
#include "cola.h"
using namespace std;
```
13. El programa deberá contar con un ciclo que le permita leer la expresión infija a convertir, leyendo cada elemento de la expresión como una cadena de entrada (arreglo de char).
14. Una vez leído cada elemento deberá ser identificado como alguno de los operadores o como un término, ya sea literal o numérico.

15. La lectura de la expresión algebraica de entrada concluye con el símbolo ;.
16. El programa deberá convertir la expresión infija a su equivalente en la notación postfija, para su funcionamiento deberá usar una pila, que va a contener los operadores pendientes de utilizar y los paréntesis izquierdos, el siguiente algoritmo indica como proceder:

INICIO

Crear pila y la cola de salida, inicialmente vacías.

Leer cada elemento de la expresión infija de entrada.

MIENTRAS entrada no esté vacía (símbolo ;) HACER

Obtener un elemento de la entrada (lo llamaremos E)

SEGUN-SEA E

CASO E es número:

Insertar E al final de la cola de salida

CASO E es la variable x:

Insertar E al final de la cola de salida

CASO E es un paréntesis izquierdo:

Insertar E en la pila

CASO E es un paréntesis derecho:

MIENTRAS en la cima no aparezca un paréntesis izquierdo HACER

Extraer elemento de la pila

Insertarlo al final de la cola de salida

FIN-MIENTRAS

SI Encontramos el paréntesis izquierdo ENTONCES

Extraerlo de la pila y destruirlo

FIN-SI

Descartar E

CASO E es un operador:

MIENTRAS La pila no este vacía y su cima sea un operador de precedencia mayor o igual que la de E

HACER

Extraer elemento de la pila

Insertarlo al final de la cola de salida

FIN-MIENTRAS

Insertar E en la pila

FIN-SEGUN-SEA

FIN-MIENTRAS

MIENTRAS Pila no esté vacía HACER

Extraer elemento de la pila

Insertarlo al final de la cola de salida

FIN-MIENTRAS

Destruir pila

FIN

17. Guárdelo con el nombre de **ejercicio 17**, compílelo, ejecútelo y pruebe si funciona correctamente con la siguiente expresión infija: $X - A * (B + (C + D)/E)$, la salida que deberá obtener es: $X A B C D + E / + * -$.
18. Fin de la Práctica.