Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI)

Actividad de aprendizaje 4:
Aplicación de Pila y Cola

Víctor Agustín Díaz Méndez
Ingeniería en Informática

Estructura de Datos I (Sección D12)
Profesor: Dr. Gutierrez Hernandez Alfredo

Problema planteado

Implemente un programa que reciba una cadena que contenga una expresión con notación infija y la convierta a su equivalente expresión con notación posfija, e imprima el resultado.

Requerimientos

a) El estilo de programación debe ser Orientado a Objetos
b) Las clases *Pila* y *Cola* utilizan un arreglo
c) La cadena con la expresión infija se pasará a una Cola
d) La conversión de infija a posfija debe hacerse con el método que utiliza una Pila
e) La expresión resultante se pasará directamente a una Cola
f) Los operadores a considerar son sólo binarios: suma (+), resta (-), multiplicación (*), división (/), y potencia (^)
g) Se incluye el manejo de paréntesis como agrupador

Resumen

Realizar esta practica pareciera un poco complicado por la “compleja” lógica que requeriría comprender el problema de pasar un tipo de dato a otro, pero si se lee los documentos suministrados por el profesor el problema seria rápido de realizar ya que ahí viene solucionada la mayor parte de la lógica de la aplicación.

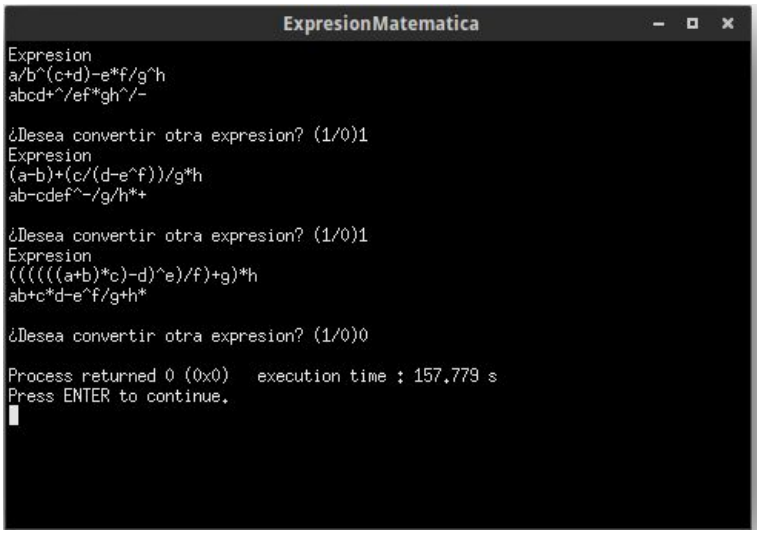
Para la entrada de datos use una variable tipo string que contendría la expresión infija a posfija, para obtener cada uno de sus caracteres utilizaría un *for* que recorrería la cadena como un arreglo mientras su contador *i* (inicializado en cero) fuera menor al tamaño de la cadena. Para obtener cada carácter use el método *.at* de la clase string lo que me permite recorrer el string de manera similar a un arreglo, como si el método *.at* fueran los corchetes del arreglo.

```
miString.at(0) //Obtiene el contenido de la posicion 0 de la cadena
```

Obteniendo cada uno de los caracteres hice métodos para procesar cada tipo de carácter pudiera ser, si era un operando, un operador, paréntesis de cierre o de apertura, o la precedencia de los operadores.

Respecto a la lógica de cada uno de los casos, solo tuve que hacer lo que estaba especificado en los documentos.

Ejecución satisfactoria:



A / B ^ (C + D) – E * F / G ^ H

R: A B C D + ^ / E F * G H ^ / –

(A - B) + (C / (D – E ^ F)) / G * H

R: A B – C D E F ^ – / G / H * +

((((((A + B) * C) – D) ^ E) / F) + G) * H

R: A B + C * D – E ^ F / G + H *