

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Lenguajes Formales y de Programación
Segundo Semestre 2020



Catedráticos: Inga. Zulma Aguirre, Ing. David Estuardo Morales, Ing. Otto Rodriguez

Tutores académicos: Daniel Urias, Luis Manuel Morales, José Manuel Véliz, Javier Cabrera

Stack Calc

Práctica #1

Tabla de contenido:

Objetivos	2
Generales:	2
Específicos:	2
Descripción:	2
Notaciones:	3
Notación Prefija:	3
Notación Infija:	3
Notación Postfija:	3
Ejemplo de evaluación de una operación en notación postfija:	4
Características del Programa:	6
Opción Cargar Archivo:	6
Estructura de los archivos de entrada:	7
Palabras reservadas:	7
Funciones:	7
Operadores Aritméticos:	8
Notaciones:	8
Opción Graficar Operación:	8
ENTREGABLES	9
RESTRICCIONES	9
ENTREGA	9

Objetivos

Generales:

- Poner en práctica los conocimientos adquiridos en la clase y laboratorio sobre paradigmas de programación.

Específicos:

- Que el estudiante sea capaz de identificar qué paradigmas se aplican de mejor manera para resolver problemas específicos de una manera simple y efectiva.
- Que el estudiante ponga en práctica conocimientos adquiridos sobre la lectura y manejo de archivos.
- Que el estudiante se familiarice con la sintaxis utilizada en el lenguaje de programación Python.

Descripción:

Con el fin de entender y profundizar sobre el tema de los paradigmas de programación y sobre los lenguajes de programación, específicamente [Python](#); se le pide al estudiante que desarrolle un programa de consola en Python el cual debe ser capaz de leer un archivo de entrada, clasificar el tipo de notación de la entrada y resolver la operación solicitada.

Los tipos de notación que se pueden utilizar en las operaciones son tres: notación polaca (prefija), notación polaca inversa (postfija) y notación infija. Se emplea la notación polaca en temas como álgebra, lógica y computación con el objetivo de eliminar los paréntesis y signos de agrupación sin perder la precedencia de los operadores.

El software a desarrollar deberá de contar con tres menús, los cuales deben de ser Cargar Archivo, Graficar Operación y Salir, la funcionalidad de cada uno de estos menús se detalla más adelante.

Para realizar los gráficos se le solicita, el estudiante deberá hacer uso de la herramienta Graphviz, cabe mencionar que puede hacer uso de la herramienta oficial o de la [librería](#) de graphviz en python.

Notaciones:

Notación Prefija:

El nombre original de esta notación es 'Notación Polaca' pero también conocida como notación prefija y es una forma de notación para la lógica, la aritmética, el álgebra y la computación. Su característica distintiva es que coloca los operadores a la izquierda de sus operandos. El resultado es una sintaxis que carece de paréntesis u otros signos de agrupación, y que puede ser analizada sin ambigüedad.

Ejemplos:

$$12 + 22 \rightarrow + 12 22$$

$$5 * 10 / (2 + 3) \rightarrow / * 5 10 + 2 3$$

$$(9 + 2) * \text{Pow}[2, 3] \rightarrow * + 9 2 \text{Pow}[2, 3]$$

Notación Infija:

Esta notación es la más común en las expresiones matemáticas, en la cual se escribe los operadores entre los operandos. Uno de los inconvenientes que presenta esta notación es la utilización de paréntesis para indicar el orden de operación.

Ejemplos:

$$\text{Fact}[6] * 9 - (2 * 3)$$

$$5 * (1 + 2 + 3) - 1$$

$$\text{Pow}(2, 3) * (2 + 3) / (15 + 1)$$

Notación Postfija:

La notación postfija también es conocida como notación polaca inversa, la cual es una variación de la notación polaca en la cual los operadores se colocan al final de los operandos. Al igual que la notación polaca, esta notación no presenta la necesidad de hacer uso de paréntesis para indicar un orden de operación.

Esta notación es la más utilizada al momento de evaluar una operación en computador, ya que permite de una forma muy sencilla y eficiente evaluar expresiones matemáticas mediante el uso de una pila (**stack**).

Ejemplos:

$12 + 22 \rightarrow 12 \ 22 \ +$

$(5 - 3) * \text{Fib}[2] \rightarrow 5 \ 3 - \text{Fib}[2] *$

$\text{Pow}[2, 4] * 5 / (7 - 2) \rightarrow \text{Pow}[2, 4] \ 5 * 7 \ 2 - /$

Ejemplo de evaluación de una operación en notación postfija:

Expresión a evaluar: $4 * \text{Pow}[2, 2] / (4.4 + 3.6)$

Representación en postfijo: $4 \ \text{Pow}[2, 2] * \ 4.4 \ 3.6 + /$

Procedimiento: para resolver una operación en notación postfija es necesario seguir los siguientes pasos:

1. Se recorre la entrada de izquierda a derecha.
2. Cuando se lee un operando se ingresa a la pila.
3. Cuando se lee un operador se sacan los últimos 2 valores ingresados a la pila, se realiza la operación que el operador indica y el resultado se ingresa en el tope de la pila.

Ejemplo:

Pila	Entrada	Acción
	4 Pow[2, 2] * 4.4 3.6 + /	Se lee el operando 4
4	Pow[2, 2] * 4.4 3.6 + /	Se lee el operando Pow[2, 2]
4 Pow[2, 2]	* 4.4 3.6 + /	Se lee el operador * op2 = Pop() → Pow[2,2] op1 = Pop() → 4 Push(op1 * op2)
16	4.4 3.6 + /	Se lee el operando 4.4
16 4.4	3.6 + /	Se lee el operando 3.6
16 4.4 3.6	+ /	Se lee el operador + op2 = Pop() → 3.6 op1 = Pop() → 4.4 Push(op1 + op2)
16 8	/	Se lee el operador / op2 = Pop() → 8 op1 = Pop() → 16 Push(op1 / op2)
2		

Al finalizar de recorrer toda la entrada se saca el último elemento de la pila, el cual representa el resultado final.

Características del Programa:

Como se menciona en el apartado anterior, solamente se contará con tres menús, los cuales se deben de mostrar en la pantalla de inicio, en esta pantalla de inicio se debe de indicar el nombre y sección del curso y nombre y carné del estudiante, de igual manera, es en esta pantalla en la cual se le pide al usuario la opción a ingresar.

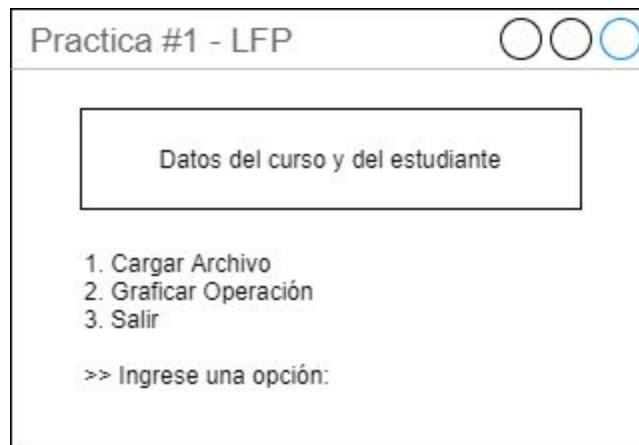


Figura #1: Prototipo de la pantalla de inicio.

Opción Cargar Archivo:

Al momento que el usuario seleccione esta opción se debe de solicitar la ruta del archivo, el cual debe de tener extensión '.lfp' y si la ruta es válida se debe leer todo el archivo, realizar todas las operación sin importar la notación que se utilice y mostrar el resultado de cada operación. La operación a realizar se debe de mostrar en la consola así como su equivalente a notación postfija, los resultados de las operación se deben de desplegar abajo de la operación realizada.

Ejemplo:

>> Operación: $(8+4) \times 2 \rightarrow$ Post: $8\ 4\ +\ 2\ *$

>> Resultado : 24

>> Operación: $4\ 5\ *\ 4\ 6\ +\ /$

>> Resultado: 2

>> Operación: $*\ +\ 2\ 3\ 4 \rightarrow$ Post: $2\ 3\ +\ 4\ *$

>> Resultado: 20

Al momento de finalizar las operaciones de un archivo, se debe de realizar una espera en consola hasta que el usuario ingrese cualquier tecla, esto con el objetivo de visualizar los resultados. Al momento que el usuario ingrese cualquier tecla, se debe de regresar a la pantalla principal.

Estructura de los archivos de entrada:

En los archivos de entrada, vendrán tanto los operadores como los operandos separados por un espacio, al inicio de cada operación se encuentran las iniciales de la notación de la operación (IN, POST, PRE) seguido de dos puntos ':' por último la operación a realizar. Solo vendrá una operación por línea. es decir, las operaciones están separadas por salto de línea.

Ejemplo:

IN: 45 + 78 * 12

POST: 45 2 * 1.5 +

PRE: + 45 78 * 2

.

.

.

IN: 15 + Fact[5] * 2

Palabras reservadas:

Las palabras reservadas del lenguaje tanto para las funciones como las operaciones son las siguientes:

Funciones:

- Factorial: Fact[n] → donde $n > 0$ y entero.
- Potencia: Pow[b, e] → donde 'b' representa la base y 'e' el exponente.
- Raíz Cuadrada: Sqrt[n] → donde $n > 0$ y entero.

Operadores Aritméticos:

- Suma: +
- Resta: -
- Multiplicación: *
- División: /
- Paréntesis: ()

Notaciones:

- Prefija: PRE
- Infija: IN
- Postfija: POST

Opción Graficar Operación:

Al momento que el usuario seleccione esta opción, se debe de pedir como entrada la operación a evaluar, esta operación siempre será ingresada en notación infija e internamente se debe de realizar la conversión a notación postfija y evaluar. Al momento de finalizar la operación se debe mostrar el resultado en consola y generar un gráfico (el nombre queda a elección del estudiante), este gráfico debe de ser generado con graphviz. Para regresar a la pantalla principal estando en esta opción, el usuario deberá de ingresar el comando '**exit**'.

Ejemplo del gráfico a generar:

Entrada: $4 * \text{Pow}[2, 2] / (4.4 + 3.6)$

Representación en postfijo: $4 \text{ Pow}[2, 2] * 4.4 3.6 + /$

Gráfico:

			Operador: *			Operador: +	Operador: /	
						3.6		
		Pow[2, 2]		4.4		4.4	8	
	4	4	16	16	16	16	16	2

ENTREGABLES

- Manual Técnico, debe incluir el código de los algoritmos empleados en la práctica y todos los aspectos para la modificación del software.
- Código fuente de la aplicación funcional, en el cual el código se debe de poder verificar los paradigmas utilizados los cuales deberá ser capaz de explicar para comprobar la autoría de su aplicación.
- Manual de usuario.

RESTRICCIONES

- La práctica debe ser implementada utilizando el lenguaje de programación Python.
- La práctica es individual.
- Las copias de prácticas tendrán automáticamente nota de 0 puntos y se reportará a los involucrados a la Escuela de Ingeniería en Ciencias y Sistemas.
- No se recibirán prácticas después de la fecha de entrega.

ENTREGA

- Fecha de entrega: Miércoles 02 de septiembre de 2020 hasta las 11:59 p.m.
- La entrega será por medio de UEDI en la actividad correspondiente.
- Tomar en cuenta que NO habrá prórroga, por lo que se solicita distribuir adecuadamente su tiempo para la elaboración de la práctica.
- Para subir la práctica deberán crear un archivo comprimido con todos los entregables detallados anteriormente. El archivo deberá tener el siguiente formato:

[LFP]Practica1_#CARNE.zip