Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 – Programación como Herramienta para la Ingeniería (II/2017)

Actividad 3

Objetivos

- Usar estructuras de datos convenientemente para evaluar una expresión matemática escrita en notación *postfix*.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** GitHub
- **Hora:** 16:55
- **Desarrollo en parejas**

Recomendación

Lean la actividad completa antes de comenzar a trabajar.

Introducción

La notación *infix* es la notación matemática que generalmente se usa. En ella, los operadores se escriben entre sus operandos. Existen otras notaciones. Una de ellas es la notación *postfix*, que ha sido muy usada por Hewlett-Packard para hacer más eficientes algunos cálculos, en la que se escriben los operadores después de sus operandos. Por ejemplo, lo siguiente es la misma expresión escrita usando ambas notaciones:

$$\textit{infix}\colon\ 12 + (3 \times 3.14)$$
$$\textit{postfix}\colon\ 1233.14 \times +$$

Como pueden notar, en la expresión en *postfix* no se puede determinar cuáles son los operandos (e.g., la multiplicación podría interpretarse como 3.1×4 y la suma, como $(3.1 \times 4) + 123$. Por esta razón, es necesario manejar el concepto de *tokens* (o usar paréntesis, pero no serán utilizados para la notación *postfix* en esta actividad).

Los *tokens*, en este contexto, son los elementos de la expresión matemática: los operadores y los operandos. En el ejemplo en *postfix* anterior ($1233.14 \times +$), hay cuatro *tokens*: 12, 3, 3.14, $\times$ y $+$.

Instrucciones

En esta actividad, deberán implementar un algoritmo que, dados los *tokens* ordenados de una expresión en *postfix*, pueda calcular el valor de tal expresión. Para lograrlo, tendrán que apoyarse particularmente en colas (*queues*) y pilas (*stacks*).

Su algoritmo deberá soportar el cálculo de expresiones que:

- Tengan largo, cantidad de operaciones y cantidad de *tokens* indefinidos.
- No tengan paréntesis.
- Solo tengan números reales como operandos.
- Solo tengan operaciones que pertenezcan a cualquier subconjunto del conjunto de operaciones de suma, resta, multiplicación, división, potencia y factorial.

En su programa, por ejemplo, pueden implementar este algoritmo mediante una función que reciba una lista (o cola) de *tokens* y que evalúe la expresión postfix compuesta de aquellos *tokens*, llamada *evaluate_postfix(tokens)*. A continuación se muestran ejemplos de su uso:

```
evaluate_postfix(["12", "3", "3.14", "*", "+"])  # Retorna 21.42
evaluate_postfix(["4", "2", "**", "2", "3", "**", "/"])  # Retorna 2
evaluate_postfix(["4", "!", "1", "!", "3", "!", "*", "/"])  # Retorna 4
evaluate_postfix(["42", "-33", "-"])  # Retorna 75
```

Pueden usar los símbolos que quieran para representar las operaciones, siempre y cuando usen **al menos dos símbolos para cada operación** (por ejemplo, "-" y "menos" para representar la resta) y los expliciten en un diccionario que relacione cada símbolo con la función que representa. Este diccionario les servirá para realizar las operaciones necesarias en su implementación; porque dado un símbolo, se puede obtener una función que realice la operación correcta, mediante el uso del diccionario.

Módulos de Python útiles

1. Es recomendable que importes el módulo math, para que así puedas utilizar la función factorial.
2. Puedes hacer uso del módulo operator, que implementa operadores en forma de funciones.

Requerimientos adicionales

Hoy e **individualmente**, respondan este form, para así contar con sus opiniones sobre la actividad. Esto es obligatorio, pues nos ayudará mucho a mejorar el curso. Ya no es necesario que pongan un comentario al principio de su código, pues el form lo reemplaza.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

"Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad."

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por "copia" o "plagio" se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.