

## C04: Pasar de forma infija a postfija

Estructuras de Datos  
Facultad de Informática - UCM

Este ejercicio consta de una única entrega, que debe enviarse al problema *DOMjudge* con identificador C04 antes del **jueves 27 de febrero a las 23:55**.

La entrega consiste en un único fichero `.cpp` que se subirá a *DOMjudge*. Puedes subir tantos intentos como quieras. Se tendrá en cuenta el último intento con el veredicto **CORRECT** que se haya realizado antes de la hora de entrega por parte de alguno de los miembros del grupo.

No olvides poner el nombre de los componentes del grupo en el fichero `.cpp`. Solo es necesario que uno de los componentes del grupo realice la entrega.

**Evaluación:** Este ejercicio se puntuará de 0 a 10. Para poder obtener una calificación superior a 0 es necesario obtener un veredicto **CORRECT**.

En este ejercicio debes realizar un programa que convierta expresiones aritméticas en forma infija a expresiones aritméticas en forma postfija. Suponemos que las expresiones en forma infija vienen dadas por la siguiente gramática:

$$\begin{aligned} e &::= e_1 + e_2 \mid e_1 * e_2 \mid (e) \mid num \\ num &::= 1 \mid 2 \mid \dots \mid 9 \end{aligned}$$

Es decir, el programa recibirá por la entrada expresiones aritméticas formadas por sumas y/o multiplicaciones de números entre 1 y 9. Por ejemplo:  $2*5+4$  y  $5*(1+4)$ . Para estas dos expresiones, el programa que realices deberá producir la salida  $25*4+$  y  $514*+$ , respectivamente.

En las expresiones en forma infija suponemos que la multiplicación tiene más prioridad que la suma, por lo que la expresión  $5*4+2$  debe entenderse como  $(5*4)+2$ . También supondremos que los operadores suma y multiplicación son asociativos por la izquierda. Esto significa que la expresión  $3+4+7$  debe entenderse como  $(3+4)+7$  y, por tanto, debe traducirse a la expresión en forma postfija  $34+7+$ . No obstante, la expresión de entrada puede contener paréntesis que alteren estas reglas de prioridad y asociatividad. Por ejemplo, si la expresión de entrada es  $3+(4+7)$ , su correspondiente en forma postfija es  $347++$ . Recuerda que las expresiones en forma postfija nunca tienen paréntesis.

Tu trabajo consiste en implementar la siguiente función:

```
Queue<char> infix_to_postfix(Queue<char> &infix);
```

Esta función recibe una cola con los caracteres de una expresión en forma infija. Por ejemplo, si la expresión de entrada es  $(2+5)*(4+3)+5$ , la cola `infix` recibida como parámetro tendrá el carácter `'` como primer elemento, el carácter `'2'` como segundo elemento, etc. La función `infix_to_postfix` debe devolver otra cola que contenga los caracteres de la expresión en forma postfija.

Para implementar esta función debes tener en cuenta las siguientes normas:

- De la cola `infix` recibida como parámetro, solamente pueden utilizarse los métodos `front()`, `pop()` y `empty()` para extraer los caracteres de la expresión de entrada.
- Con respecto a la cola a devolver por la función, solamente puede utilizarse el método `push()` para introducir los caracteres de la expresión en forma postfija.
- Puede utilizarse una pila de elementos `char` para almacenar temporalmente algunos de los caracteres de la expresión de entrada.

- Aparte de la cola de entrada, la cola de salida y la pila, **no** puede utilizarse ninguna otra estructura de datos adicional (array, lista, etc.) para almacenar los caracteres de la expresión de entrada.

*Indicación:* Tan solo es necesario almacenar caracteres '+', '\*' y '(' en la pila de char.

Utiliza la plantilla que encontrarás en el Campus Virtual.

## Entrada

La entrada contiene una serie de casos de prueba, uno por línea. Cada caso de prueba consiste en una expresión en forma infija como las descritas anteriormente. La expresión no contiene espacios.

## Salida

Para cada caso se escribirá una línea con la misma expresión de entrada, pero expresada en forma postfija.

## Entrada de ejemplo

```
3
3+7
3+7+2
3+(7+2)
3*7+2
3+7*2
(3+7)*2
3*(7+2)
```

## Salida de ejemplo

```
3
37+
37+2+
372++
37*2+
372*+
37+2*
372+*
```