

## Evaluando expresiones en notación posfija

El objetivo de este control es resolver un problema utilizando TADs lineales vistos en clase.

### 1) El problema

Una expresión aritmética construida con los operadores binarios  $+$ ,  $-$ ,  $*$  y  $/$  y operandos (números de un único dígito) se dice que está en forma posfija si es o bien un sólo operando o dos expresiones en forma posfija, una tras otra, seguidas inmediatamente de un operador. A continuación puedes ver un ejemplo de una expresión escrita en notación infija habitual, junto con su forma posfija:

Forma infija:  $(8 / (2 - 1)) * (8 + 2)$

Forma posfija:  $821-/82+*$

Implementa un algoritmo iterativo que devuelva el valor resultante de evaluar una expresión dada en forma posfija. Para ahorrar aspectos que compliquen el algoritmo se supone que la expresión no contiene operadores unarios ni potencias y que, como se ha dicho antes, los operandos tienen un único dígito.

Ejemplo de entrada / salida (en la entrada pueden verse tres expresiones en notación posfija y en la salida el valor resultado de evaluar cada expresión de su izquierda):

Entrada	Salida
5	5
55-	0
53-2+	4

### 2) Trabajo a realizar

Se proporciona el archivo `main.cpp` en el que se implementa toda la lógica de entrada / salida necesaria.

Hay que añadir a dicho archivo la implementación de la función

```
int evaluar(string exp);
```

que devuelva el resultado de evaluar la expresión en notación posfija recibida en la cadena `exp`, así como el resto de definiciones y directivas que se consideren oportunas, siempre y cuando no se modifique la función `main`.

La cadena `exp` contiene una expresión en notación posfija bien formada, con operandos de un único dígito, operadores binarios ( $+$ ,  $-$ ,  $*$  y  $/$  representando la suma, la resta, la multiplicación y la división, respectivamente) y sin ningún espacio en blanco.

La solución deberá utilizar también los TADs vistos en clase que se consideren oportunos (úsense las implementaciones basadas en listas enlazadas).