

```

{
    NodoPila* n;
    while(!pilaVacía())
    {
        n = cima;
        cima = cima -> siguiente;
        delete n;
    }
}

```

11.4. EVALUACIÓN DE EXPRESIONES ARITMÉTICAS CON PILAS

Una *expresión aritmética* está formada por operandos y operadores. La expresión $x * y - (a + b)$ consta de los operadores $*$, $-$, $+$ y de los operandos x , y , a , b . Los operandos pueden ser valores constantes, variables o, incluso, otra expresión. Los operadores son los símbolos conocidos de las operaciones matemáticas.

La evaluación de una expresión aritmética da lugar a un valor numérico, se realiza sustituyendo los operandos que son variables por valores concretos y ejecutando las operaciones aritméticas representadas por los operadores. Si los operandos de la expresión anterior toman los valores: $x = 5$, $y = 2$, $a = 3$, $b = 4$ el resultado de la evaluación es:

$$5 * 2 - (3 + 4) = 5 * 2 - 7 = 10 - 7 = 3$$

La forma habitual de escribir expresiones matemáticas sitúa el operador entre sus dos operandos. La expresión anterior está escrita de esa forma, recibe el nombre de *notación infija*. Esta forma de escribir las expresiones exige, en algunas ocasiones, el uso de paréntesis para *encerrar* subexpresiones con mayor prioridad.

Los operadores, como es sabido, tienen distintos niveles de precedencia o prioridad a la hora de su evaluación. A continuación, se recuerda estos niveles de prioridad en orden de mayor a menor:

Paréntesis	: ()
Potencia	: ^
Multiplicación/división	: *, /
Suma/Resta	: +, -

Normalmente, en una expresión hay operadores con la misma prioridad, a igualdad de precedencia, los operadores se evalúan de izquierda a derecha (*asociatividad*), excepto la potencia que es de derecha a izquierda.

11.4.1. Notación *prefija* y notación *postfija* de una expresiones aritmética

Las operaciones aritméticas escritas en *notación infija* en muchas ocasiones necesitan usar paréntesis para indicar el orden de evaluación. Las expresiones

$$r = a * b / (a + c) \qquad g = a * b / a + c$$

son distintas al no poner paréntesis en la expresión g . Igual ocurre con estas otras:

$$r = (a - b) ^ c + d \qquad g = a - b ^ c + d$$

Existen otras formas de escribir expresiones aritméticas, que se diferencian por la ubicación del operador respecto de los operandos. La notación en la que el operador se coloca delante de los dos operandos, *notación prefija*, se conoce también como *notación polaca* por el matemático polaco que la propuso. En el Ejemplo 11.4 se escriben expresiones en notación *prefija* o notación polaca.

EJEMPLO 11.4. Dadas las expresiones: $a * b / (a + c)$; $a * b / a + c$; $(a - b)^c + d$. Escribir las expresiones equivalentes en notación prefija.

Paso a paso, se escribe la transformación de cada expresión algebraica en la expresión equivalente en notación polaca.

$$\begin{aligned} a * b / (a + c) \text{ (infija)} &\rightarrow a * b / + ac \rightarrow * ab / + ac \rightarrow / *ab + ac \text{ (polaca)} \\ a * b / a + c \text{ (infija)} &\rightarrow * ab / a + c \rightarrow / * aba + c \rightarrow + / *abac \text{ (polaca)} \\ (a - b)^c + d \text{ (infija)} &\rightarrow -ab^c + d \rightarrow ^-abc + d \rightarrow +^-abcd \text{ (polaca)} \end{aligned}$$

Nota

La propiedad fundamental de la notación polaca es que el orden de ejecución de las operaciones está determinado por las posiciones de los operadores y los operandos en la expresión. No son necesarios los paréntesis al escribir la expresión en notación polaca, como se observa en el Ejemplo 11.4.

Notación postfija

Hay más formas de escribir las expresiones. La notación *postfija* o *polaca inversa* coloca el operador a continuación de sus dos operandos.

EJEMPLO 11.5. Dadas las expresiones: $a*b/(a+c)$; $a*b/a+c$; $(a-b)^c+d$. Escribir las expresiones equivalentes en notación postfija.

Paso a paso se transforma cada subexpresión en notación polaca inversa.

$$\begin{aligned} a*b/(a+c) \text{ (infija)} &\rightarrow a*b/ac+ \rightarrow ab*/ac+ \rightarrow ab*ac+/ \quad \text{(polaca inversa)} \\ a*b/a+c \text{ (infija)} &\rightarrow ab*/a+c \rightarrow ab*a/+c \rightarrow ab*a/c+ \quad \text{(polaca inversa)} \\ (a-b)^c+d \text{ (infija)} &\rightarrow ab-^c+d \rightarrow ab-c^+d \rightarrow ab-c^+d+ \quad \text{(polaca inversa)} \end{aligned}$$

Recordar

Las diferentes formas de escribir una misma expresión algebraica dependen de la ubicación de los operadores respecto a los operandos. Es importante tener en cuenta que tanto en la notación prefija como en la postfija no son necesarios los paréntesis para cambiar el orden de evaluación.

11.4.2. Evaluación de una expresión aritmética

La evaluación de una expresión aritmética escrita de *manera habitual*, en *notación infija*, se realiza en dos pasos principales:

- 1.º Transformar la expresión de notación infija a postfija.
- 2.º Evaluar la expresión en notación postfija.

El *TAD Pila* es fundamental en los algoritmos que se aplican a cada uno de los pasos. El orden que fija la estructura pila asegura que el *último en entrar es el primero en salir*, y de esa forma el algoritmo de transformación a *postfija* sitúa los operadores después de sus operandos, con la prioridad o precedencia que le corresponde. Una vez que se tiene la expresión en notación *postfija*, se utiliza otra pila, de elementos numéricos, para guardar los valores de los operandos, y de las operaciones parciales con el fin de obtener el valor numérico de la expresión.

11.4.3. Transformación de una expresión infija a *postfija*

Se parte de una expresión en *notación infija* que tiene operandos, operadores y puede tener paréntesis. Los operandos se representan con letras, los operadores son éstos:

^ (potenciación), *, /, +, - .

La transformación se realiza utilizando una pila para guardar operadores y los paréntesis izquierdos. La expresión aritmética se lee del teclado y se procesa carácter a carácter. Los operandos pasan directamente a formar parte de la expresión en *postfija* la cual se guarda en un array. Un operador se mete en la pila si se cumple que:

- La pila está vacía, o,
- El operador tiene mayor prioridad que el operador cima de la pila, o bien,
- El operador tiene igual prioridad que el operador cima de la pila y se trata de la máxima prioridad.

Si la prioridad es menor o igual que la de *cima pila*, se saca el elemento cima de la pila, se pone en la expresión en *postfija* y se vuelve a hacer la comparación con el nuevo elemento cima.

El paréntesis izquierdo siempre se mete en la pila; ya en la pila se les considera de mínima prioridad para que todo operador que se encuentra dentro del paréntesis entre en la pila. Cuando se lee un paréntesis derecho se sacan todos los operadores de la pila y pasan a la expresión *postfija*, hasta llegar a un paréntesis izquierdo que se elimina ya que los paréntesis no forman parte de la expresión *postfija*. El algoritmo termina cuando no hay más ítems de la expresión origen y la pila está vacía.

Por ejemplo, dada la expresión $a * (b + c - (d / e^f) - g) - h$ escrita en *notación infija*, a continuación, se va a ir formando, paso a paso, la expresión equivalente en *postfija*.

Expresión en *postfija*

- | | |
|----|--|
| a | Operando a pasa a la expresión <i>postfija</i> ; operador * a la pila. |
| ab | Operador (pasa a la pila; operando b a la expresión. |

abc

Operador + pasa a la pila; operando a la expresión.

En este punto, el estado de la pila:

+
(
*

El siguiente carácter de la expresión, -, tiene igual prioridad que el operador de la cima (+), da lugar:

-
(
*

abc+

El operador (se mete en la pila; el operando d a la expresión.

abc+d

El operador / pasa a la pila; el operando e a la expresión.

abc+de

El operador ^ pasa a la pila; el operando f a la expresión.

abc+def

El siguiente item,) (paréntesis derecho), produce que se vacíe la pila hasta un (. La pila, en este momento, dispone de estos operadores:

^
/
(
-
)
*

abc+def^/

El algoritmo saca operadores de la pila hasta un '(' y da lugar a la pila:

-
(
*

abc+def^/-

El operador - pasa a la pila y, a su vez, se extrae -; el siguiente carácter, el operando g pasa a la expresión.

abc+def^/-g

El siguiente carácter es), por lo que son extraídos de la pila los operadores hasta un (, la pila queda de la siguiente forma:

*

abc+def^/-g-

El siguiente carácter es el operador -, hace que se saque de la pila el operador * y se meta en la pila el operador -.

abc+def^/-g-*

Por último, el operando h pasa directamente a la expresión.

abc+def^/-g-*h Fin de entrada, se vacía la pila pasando los operadores a la expresión:
abc+def^/-g-*h-

El seguimiento realizado pone de manifiesto la importancia de considerar al paréntesis izquierdo un operador de mínima prioridad dentro de la pila, para que los operadores, dentro de un paréntesis, se metan en la pila y después extraerlos cuando se trata el paréntesis derecho. También tiene un comportamiento distinto el operador de potenciación dentro y fuera de la pila, debido a que tienen asociatividad de derecha a izquierda. Las prioridades se fijan en la Tabla 11.1.

Tabla 11.1. Tabla de prioridades de los operadores considerados.

Operador	Prioridad dentro pila	Prioridad fuera pila
^	3	4
*, /	2	2
+, -	1	1
(0	5

Observe, que el paréntesis derecho no se considera ya que éste provoca sacar operadores de la pila hasta el paréntesis izquierdo.

Algoritmo de paso de notación *infija* a *postfija*

Los pasos a seguir para transformar una expresión algebraica de *notación infija* a *postfija*:

1. Obtener caracteres de la expresión y repetir los pasos 2 al 4 para cada carácter.
2. Si es un operando, pasarlo a la expresión postfija.
3. Si es operador:
 - 3.1. Si la pila está vacía, meterlo en la pila. Repetir a partir de 1.
 - 3.2. Si la pila no está vacía:

Si prioridad del operador es mayor que prioridad del operador cima, meterlo en la pila y repetir a partir de 1.

Si prioridad del operador es menor o igual que prioridad del operador cima, sacar operador cima de la pila y ponerlo en la expresión postfija, volver a 3.
4. Si es paréntesis derecho:
 - 4.1. Sacar operador cima y ponerlo en la expresión postfija.
 - 4.2. Si el nuevo operador cima es paréntesis izquierdo, suprimir elemento cima.
 - 4.3. Si cima no es paréntesis izquierdo, volver a 4.1.
 - 4.4. Volver a partir de 1.
5. Si quedan elementos en la pila pasarlos a la expresión postfija.
6. Fin del algoritmo.

Codificación del algoritmo de transformación a *postfija*

Se necesita crear una pila de caracteres para guardar los operadores. Se utiliza el diseño de Pila genérica del Apartado 11.3. La expresión original se lee del teclado en una cadena de suficiente tamaño. También se declara una estructura para representar un elemento de la ex-