

Conv_postfija (EI, EPOS)

{Este algoritmo traduce una expresión infija —EI— a postfija —EPOS—, haciendo uso de una pila —PILA—. MAX es el número máximo de elementos que puede almacenar la pila}

1. Hacer TOPE $\leftarrow 0$
2. Mientras (EI sea diferente de la cadena vacía) Repetir
 - Tomar el símbolo más a la izquierda de EI. Recortar luego la expresión
 - 2.1 Si (el símbolo es paréntesis izquierdo)
 - entonces {Poner símbolo en PILA. Se asume que hay espacio en PILA}
 - Llamar a Pone con PILA, TOPE, MAX y símbolo
 - si no
 - 2.1.1 Si (el símbolo es paréntesis derecho)
 - entonces
 - 2.1.1.1 Mientras (PILA[TOPE] \neq paréntesis izquierdo) Repetir
 - Llamar a Quita con PILA, TOPE y DATO
 - Hacer EPOS \leftarrow EPOS + DATO
 - 2.1.1.2 {Fin del ciclo del paso 2.1.1.1}
 - Llamar a Quita con PILA, TOPE y DATO
 - {Se quita el paréntesis izquierdo de PILA y no se agrega a EPOS}
 - si no
 - 2.1.1.3 Si (el símbolo es un operando)
 - entonces
 - Agregar símbolo a EPOS
 - si no {Es un operador}
 - Llamar Pila_vacía con PILA, TOPE y BAND
 - 2.1.1.3A Mientras (BAND = FALSO) y (la prioridad del operador sea menor o igual que la prioridad del operador que está en la cima de PILA) Repetir
 - Llamar a Quita con PILA, TOPE y DATO
 - Hacer EPOS \leftarrow EPOS + DATO
 - Llamar a Pila_vacía con PILA, TOPE y BAND
 - 2.1.1.3B {Fin del ciclo del paso 2.1.1.3A}
 - Llamar a Pone con PILA, TOPE, MAX y símbolo
 - 2.1.1.4 {Fin del condicional del paso 2.1.1.3}
 - 2.1.2 {Fin del condicional del paso 2.1.1}
 - 2.2 {Fin del condicional del paso 2.1}
3. {Fin del ciclo del paso 2}
4. Llamar a Pila_vacía con PILA, TOPE y BAND
5. Mientras (BAND = FALSO) Repetir
 - Llamar a Quita con PILA, TOPE y DATO
 - Hacer EPOS \leftarrow EPOS + DATO
 - Llamar a Pila_vacía con PILA, TOPE y BAND
6. {Fin del ciclo del paso 5}
7. Escribir EPOS

Conv_prefija (EI, EPRE)

{Este algoritmo traduce una expresión en notación infija, EI a prefija, EPRE, haciendo uso de una pila —PILA—}

{TOPE es una variable de tipo entero y MAX representa el máximo número de elementos que puede almacenar la pila}

1. Hacer TOPE \leftarrow 0

2. Mientras (EI sea diferente de la cadena vacía) Repetir

Tomar el símbolo más a la derecha de EI recorriendo luego la expresión

2.1 Si (el símbolo es paréntesis derecho)

entonces {Poner símbolo en pila}

Llamar a Pone con PILA, TOPE, MAX y símbolo

si no

2.1.1 Si (símbolo es paréntesis izquierdo)

entonces

2.1.1.1 Mientras (PILA[TOPE] \neq paréntesis derecho) Repetir

Llamar a Quita con PILA, TOPE y DATO

Hacer EPRE \leftarrow EPRE + DATO

2.1.1.2 {Fin del ciclo del paso 2.1.1.1}

{Sacamos el paréntesis derecho de PILA y no se agrega a EPRE}

Llamar a Quita con PILA, TOPE y DATO

si no

2.1.1.3 Si (símbolo es un operando)

entonces

Agregar símbolo a EPRE

si no {Es un operador}

Llamar a Pila_vacía con PILA, TOPE y BAND

2.1.1.3A Mientras (BAND = FALSO) y (la prioridad del operador sea menor que la prioridad del operador que está en la cima de PILA) Repetir

Llamar a Quita con PILA, TOPE y DATO

Hacer EPRE \leftarrow EPRE + DATO

Llamar a Pila_vacía con PILA, TOPE y BAND

2.1.1.3B {Fin del ciclo del paso 2.1.1.3A}

Llamar a Pone con PILA, TOPE, MAX y símbolo

2.1.1.4 {Fin del condicional del paso 2.1.1.3}

2.1.2 {Fin del condicional del paso 2.1.1}

2.2 {Fin del condicional del paso 2.1}

3. {Fin del ciclo del paso 2}

Llamar a Pila_vacía con PILA, TOPE y BAND

4. Mientras (BAND = FALSO) Repetir

Llamar a Quita con PILA, TOPE y DATO

Hacer EPRE \leftarrow EPRE + DATO

Llamar a Pila_vacía con PILA, TOPE y BAND

5. {Fin del ciclo del paso 4}

6. Escribir EPRE en forma invertida