Funcionalidad del código

* El programa implementa un traductor de notación infija a postfija (también llamada notación polaca inversa).
* Entrada: Acepta expresiones matemáticas básicas con operadores +, -, \*, / y números enteros (como: 3 + 5 \* 2).
* Salida: Imprime la expresión en formato postfijo (como: 3 5 2 \* +).
* Precedencia de operadores: Respeta las reglas matemáticas (\* y / tienen mayor prioridad que + y -).

Pruebas:

|  |  |  |
| --- | --- | --- |
| input | output | observación |
| 2 + 3 \* 4 | 2 3 4 \* + | Bien detecta la multiplicación es primero que la suma |
| 10 / 2 - 3 | 10 2 / 3 - | Primero la división y despues la resta |
| 7 + 2 + | Error de sintaxis | Falta un numo después del signo ‘+’ |

Limitaciones y errores encontrados:

* No soporta paréntesis:Si se ingresa (2 + 3) \* 4, el programa no lo maneja correctamente (genera error o salida incorrecta).
* No valida expresiones incompletas:Ejemplo: 5 + produce un error de sintaxis, pero el mensaje es genérico ("Error: syntax error").
* Caracteres inválidos:Si se ingresa 2 $ 3, el lexer llama a yyerror("Carácter inválido"), pero el programa termina abruptamente.