

**Міністерство освіти і науки України
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА
ФРАНКА
Факультет прикладної математики та інформатики**

Кафедра програмування

**ЛАБОРАТОРНА РОБОТА № 5
Зворотній польський запис
з курсу “Алгоритми та структури даних”**

Виконала:

**Студентка групи ПМІ-13
Демко Сніжана Іванівна**

Львів - 2024

Мета: навчитися реалізовувати алгоритми перетворення арифметичних виразів у зворотній польський запис та обчислювати вирази у зворотньому польському записі. Покращити вміння роботи із стеками.

Зворотній польський запис - форма запису математичних виразів, в якій знаки операцій розташовано після операндів.

Виконання:

- 1) Реалізація алгоритму перетворення у зворотній польський запис:
 - Для перетворення вхідного арифметичного виразу у зворотній польський запис було використано стек.
 - Використано функцію *priority*, яка визначає пріоритет операторів.
 - Функція *toPostfix* проходить через вхідний вираз, перетворюючи його у зворотній польський запис за допомогою стеку.
- 2) Реалізація алгоритму обчислення виразу у зворотньому польському записі:
 - Для обчислення виразу у зворотньому польському записі також використовується стек.
 - Функція *calculate* отримує зворотній польський запис виразу і обчислює його, використовуючи стек для зберігання проміжних результатів.
- 3) Тести:
 - Для перевірки коректності роботи функцій було створено Google Tests.
 - Тести перевіряють різні аспекти роботи функцій: визначення пріоритету операторів, перевірка на цифри та оператори, застосування операторів, перетворення у зворотній польський запис та обчислення виразів у цьому записі.

Результати роботи програмної реалізації та тестування:

Приклад №1

```
1. Enter infix notation
2. Enter postfix notation
3. Exit
Enter your choice: 1

Enter infix notation: 4-8*7+5/6
Postfix notation: 487*-56/+
Result: -52
```

```
1. Enter infix notation
2. Enter postfix notation
3. Exit
Enter your choice: 2

Enter postfix notation: 487*-56/+
Infix notation: ((4-(8*7))+(5/6))
Result: -52

1. Enter infix notation
2. Enter postfix notation
3. Exit
Enter your choice: 3

Exiting...
```

Приклад №2

```
1. Enter infix notation
2. Enter postfix notation
3. Exit
Enter your choice: 1

Enter infix notation: (8+9-3)/(1+3*2)+5
Postfix notation: 89+3-132*/5+
Result: 7

1. Enter infix notation
2. Enter postfix notation
3. Exit
Enter your choice: 2

Enter postfix notation: 89+3-132*/5+
Infix notation: (((8+9)-3)/(1+(3*2)))+5
Result: 7

1. Enter infix notation
2. Enter postfix notation
3. Exit
Enter your choice: 3

Exiting...
[=====] Running 7 tests from 7 test cases.
[-----] Global test environment set-up.
[-----] 1 test from PriorityTest
[RUN    ] PriorityTest.OperatorPriority
[OK   ] PriorityTest.OperatorPriority (0 ms)
[-----] 1 test from PriorityTest (1 ms total)
```

```
[-----] 1 test from toInfixTest
[ RUN   ] toInfixTest.ConvetrToInfix
[   OK  ] toInfixTest.ConvetrToInfix (0 ms)
[-----] 1 test from toInfixTest (1 ms total)

[-----] 1 test from IsDigitTest
[ RUN   ] IsDigitTest.CheckIfDigit
[   OK  ] IsDigitTest.CheckIfDigit (0 ms)
[-----] 1 test from IsDigitTest (1 ms total)

[-----] 1 test from IsOperandTest
[ RUN   ] IsOperandTest.CheckIfOperand
[   OK  ] IsOperandTest.CheckIfOperand (0 ms)
[-----] 1 test from IsOperandTest (0 ms total)

[-----] 1 test from ApplyOperatorTest
[ RUN   ] ApplyOperatorTest.OperatorApplication
[   OK  ] ApplyOperatorTest.OperatorApplication (0 ms)
[-----] 1 test from ApplyOperatorTest (1 ms total)

[-----] 1 test from ToPostfixTest
[ RUN   ] ToPostfixTest.ConvertToPostfix
[   OK  ] ToPostfixTest.ConvertToPostfix (0 ms)
[-----] 1 test from ToPostfixTest (2 ms total)

[-----] 1 test from CalculateTest
[ RUN   ] CalculateTest.EvaluatePostfix
[   OK  ] CalculateTest.EvaluatePostfix (1 ms)
[-----] 1 test from CalculateTest (2 ms total)

[-----] Global test environment tear-down
[=====] 7 tests from 7 test cases ran. (28 ms total)
[ PASSED ] 7 tests.
```

Висновок: Виконавши лабораторну роботу було реалізовано алгоритми перетворення арифметичного виразу у зворотній польський запис та обчислення виразу у цьому форматі. Також було створено відповідні тести для перевірки коректності роботи функцій. В результаті у процесі виконання лабораторної роботи було покращено розуміння алгоритмів перетворення та обчислення виразів у зворотньому польському записі.