ЛАБОРАТОРНА РОБОТА 5
Обернений (постфіксний) польський запис

ТЕОРЕТИЧНИЙ МАТЕРІАЛ

Посилання: Польський інверсний запис (постфіксна нотація)

Приклади арифметичних виразів у інфіксній нотації:

$$3 + 4; \quad 5 * 2 + 3; \quad 5 * (2 + 3); \quad (8 + 9 - 3)/(1 + 3 * 2) + 5$$

Приклади цих же виразів у постфіксній нотації (обернений польський запис):

$$3\ 4\ +; \quad 5\ 2\ *\ 3\ +; \quad 5\ 2\ 3\ +\ *; \quad 8\ 9\ +\ 3\ -\ 1\ 3\ 2\ *\ +\ /\ 5\ +$$

Алгоритм побудови оберненого польського запису арифметичного виразу

Для простоти реалізації вважатимемо, що операндами і результатом арифметичного виразу є одноцифрові натуральні числа.

Припишемо знакам операцій і дужкам числові значення їхніх пріоритетів. Використаємо також ще одну хитрість: щоб однаково опрацьовувати порожній та непорожній стек, занесемо у вершину порожнього стека деякий спеціальний символ (наприклад, '#') і припишемо йому такий самий пріоритет, як і дужкам:

Символи	Пріоритет
* /	2
+ -	1
# ( )	0

Щоб побудувати постфіксний вираз (рядок[1] postfix), послідовно перебирають і аналізують символи інфіксного виразу (рядка infix):

- якщо поточний символ - цифра, то її заносять у рядок postfix;
- якщо поточний символ - знак арифметичної операції, то вилучають зі стека і заносять до рядка postfix елементи доти, доки їхній пріоритет не менший за пріоритет поточного символа; після цього знак операції додають в стек;
- якщо поточний символ - відкривна дужка, то її додають в стек;
- якщо поточний символ - закривна дужка, то до рядка postfix заносять усі символи, які є у стекові над відкривною дужкою; після цього вилучають відкривну дужку зі стека.

Після опрацювання усіх символів рядка infix вміст стека (крім спеціального символу #) переносять до postfix.

[1] Замість рядків можна використовувати масиви. Тоді обмеження на одноцифровість операндів можна зняти.

Алгоритм обчислення значення арифметичного виразу у постфіксній нотації

Польський запис переглядається посимвольно зліва направо:

- якщо поточний символ - цифра, то вона перетворюється в числовий тип і додається в стек;
- якщо поточний символ - оператор, то дістати зі стека два операнди, виконати операцію і результат занести в стек.

В кінці у стеку залишиться результат виразу.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Використавши одну з трьох реалізацій стека — на основі масивів, на основі зв'язних списків або STL (*див. лаб. роб. 2*) — реалізувати функції **побудови** та **обчислення** оберненого польського виразу, який складається з одноцифрових операндів, чотирьох арифметичних операторів (+, -, *, /) та круглих дужок. Продемонструвати роботу на тестових прикладах.

Зразок функції main() та результат її виконання:

```
int main(){
  char infix[] = "(8+9-3)/(1+3*2)+5";
  char* postfix;
  toPostfix(infix, postfix);
  cout << "Infix notation: " << infix << endl;
  cout << "Postfix notation: " << postfix << endl;
  cout << "Result: " << calcPostfix(postfix) << endl;
  delete[] postfix;
  return 0;
}
```

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