



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий

Кафедра Вычислительной техники

ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ №1,2

«Преобразование алгебраического выражения в обратную
польскую запись на Python», «Вычисления значения выражения,
записанного в обратной польской записи на Python»

по дисциплине

«Теория формальных языков»

Выполнил студент группы ИКБО-04-22

Кликушин В.И.

Принял ассистент

Боронников А.С.

Практическая работа
выполнена

«_04_»__09____2023 г.

«Зачтено»

«__»____2023 г.

Москва 2023

СОДЕРЖАНИЕ

1 ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1	3
1.1 ПОСТАНОВКА ЗАДАЧИ.....	3
1.2 КОД ПРОГРАММЫ	4
1.3 НАБОР ТЕСТОВ	5
1.4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ.....	6
2 ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2	8
2.1 ПОСТАНОВКА ЗАДАЧИ.....	8
2.2 КОД ПРОГРАММЫ	9
2.3 НАБОР ТЕСТОВ	10
2.4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ.....	11
3 ВЫВОДЫ.....	13
4 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ.....	14

1 ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1

1.1 ПОСТАНОВКА ЗАДАЧИ

Разработать программу на выбранном языке программирования для преобразования простого алгебраического выражения в обратную польскую запись.

1.2 КОД ПРОГРАММЫ

В качестве языка программирования был выбран Python.

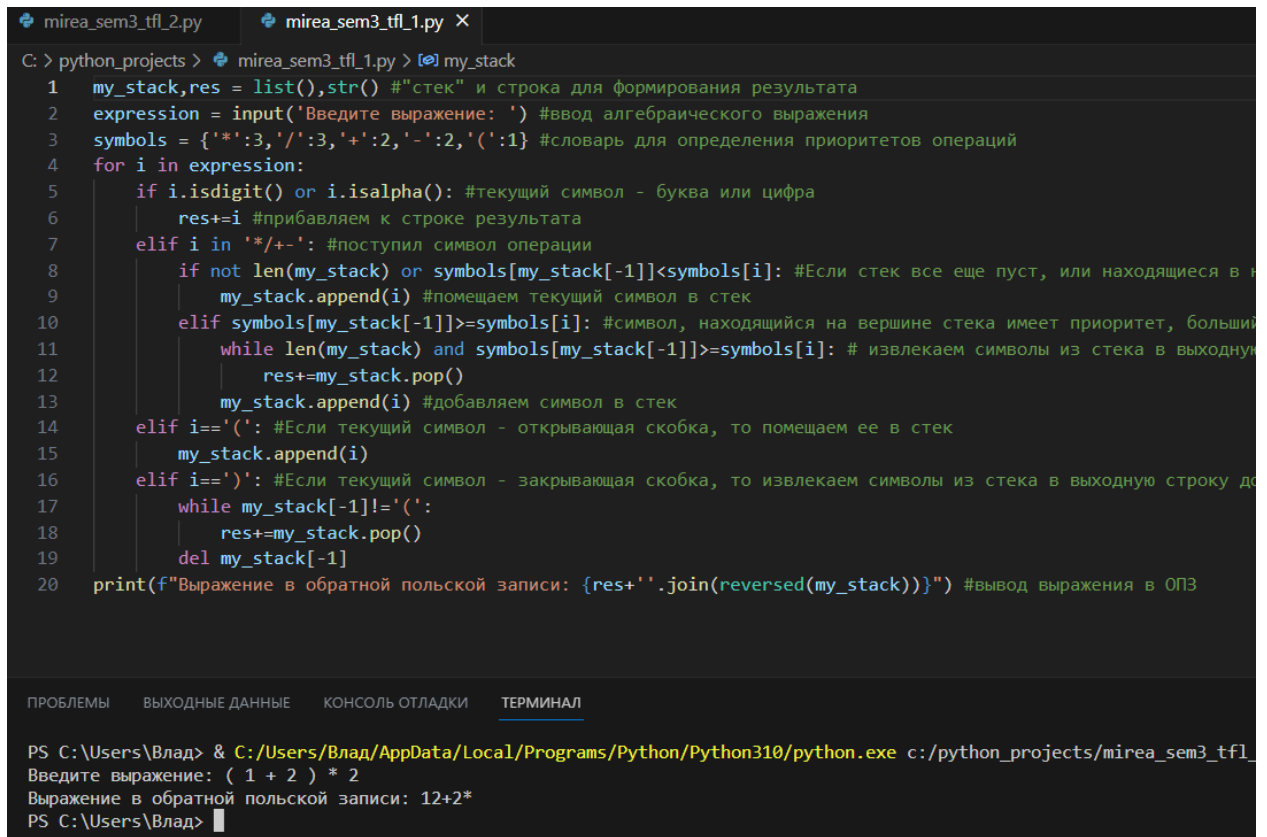
```
1 my_stack,res = list(),str() #"стек" и строка для формирования результата
2 expression = input('Введите выражение: ') #ввод алгебраического выражения
3 symbols = {'*':3,'/':3,'+':2,'-':2,'(':1} #словарь для определения
    приоритетов операций
4 for i in expression:
5     if i.isdigit() or i.isalpha(): #текущий символ - буква или цифра
6         res+=i #прибавляем к строке результата
7     elif i in '*/+-': #поступил символ операции
8         if not len(my_stack) or symbols[my_stack[-1]]<symbols[i]: #Если стек
            все еще пуст, или находящиеся в нем символы имеют меньший приоритет, чем
            приоритет текущего символа
9             my_stack.append(i) #помещаем текущий символ в стек
10            elif symbols[my_stack[-1]]>=symbols[i]: #символ, находящийся на
                вершине стека имеет приоритет, больший или равный приоритету текущего символа
11                while len(my_stack) and symbols[my_stack[-1]]>=symbols[i]: #
                    извлекаем символы из стека в выходную строку до тех пор, пока выполняется это
                    условие
12                    res+=my_stack.pop()
13                    my_stack.append(i) #добавляем символ в стек
14            elif i=='(': #Если текущий символ - открывающая скобка, то помещаем ее в
                стек
15                my_stack.append(i)
16            elif i==')': #Если текущий символ - закрывающая скобка, то извлекаем
                символы из стека в выходную строку до тех пор, пока не встретим в стеке
                открывающую скобку
17                while my_stack[-1]!='(':
18                    res+=my_stack.pop()
19                del my_stack[-1]
20 print(f"Выражение в обратной польской записи:
    {res+''.join(reversed(my_stack))}") #вывод выражения в ОПЗ
```

1.3 НАБОР ТЕСТОВ

Таблица 1. Таблица тестирования программы

Преобразование простого алгебраического выражения в обратную польскую запись		
Номер теста	Входные данные	Эталон результата
1	$(1 + 2) * 2$	12+2*
2	$(8+2*5)/(1+3*2-4)$	825*+132*+4-/
3	2+2	22+
4	$((5+7*2-1)/(2*1+1))+1$	572*+1-21*1+/1+

1.4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

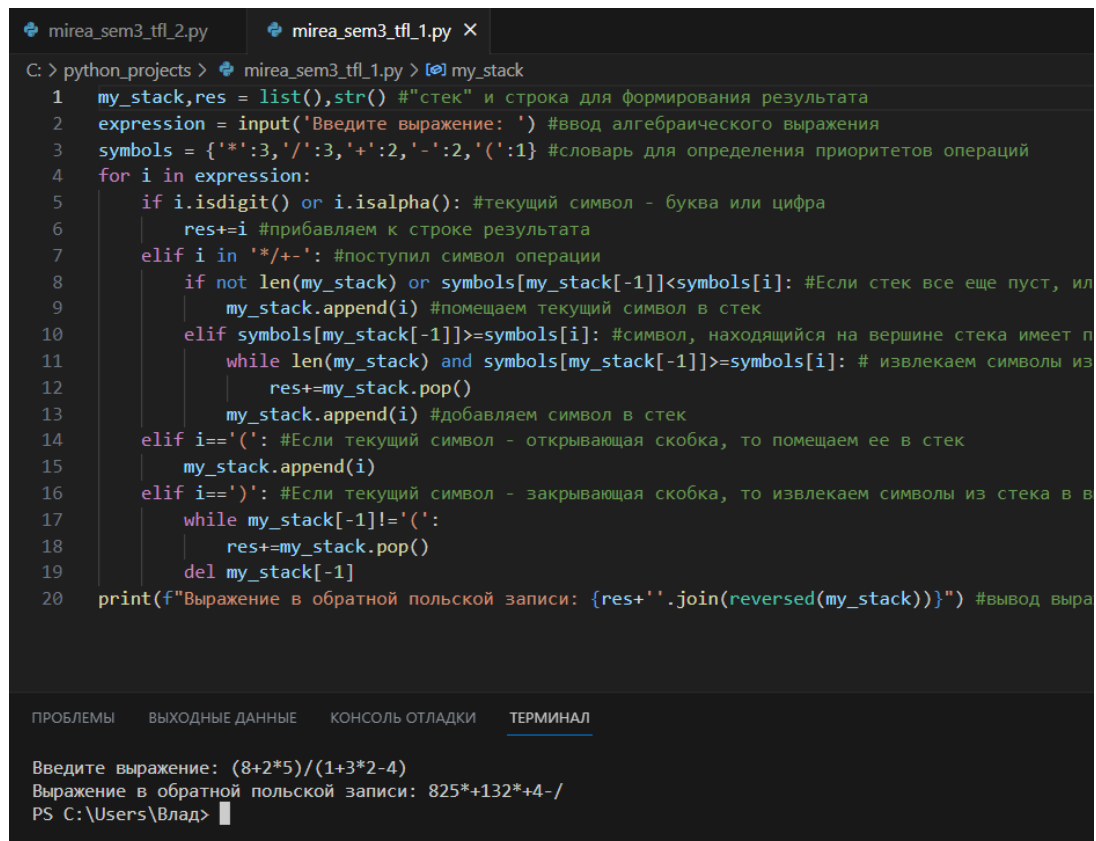


```
C: > python_projects > mirea_sem3_tfl_1.py > my_stack
1 my_stack, res = list(), str() # "стек" и строка для формирования результата
2 expression = input('Введите выражение: ') # ввод алгебраического выражения
3 symbols = {'*':3, '/':3, '+':2, '-':2, '(':1} # словарь для определения приоритетов операций
4 for i in expression:
5     if i.isdigit() or i.isalpha(): # текущий символ - буква или цифра
6         res+=i # прибавляем к строке результата
7     elif i in '+-/*': # поступил символ операции
8         if not len(my_stack) or symbols[my_stack[-1]]<symbols[i]: # Если стек все еще пуст, или находящиеся в нем
9             my_stack.append(i) # помещаем текущий символ в стек
10        elif symbols[my_stack[-1]]>=symbols[i]: # символ, находящийся на вершине стека имеет приоритет, больший
11            while len(my_stack) and symbols[my_stack[-1]]>=symbols[i]: # извлекаем символы из стека в выходную строку
12                res+=my_stack.pop()
13            my_stack.append(i) # добавляем символ в стек
14        elif i=='(': # Если текущий символ - открывающая скобка, то помещаем ее в стек
15            my_stack.append(i)
16        elif i==')': # Если текущий символ - закрывающая скобка, то извлекаем символы из стека в выходную строку до
17            while my_stack[-1]!='(':
18                res+=my_stack.pop()
19            del my_stack[-1]
20 print(f"Выражение в обратной польской записи: {res+''.join(reversed(my_stack))}") # вывод выражения в ОПЗ
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

PS C:\Users\Влад> & C:/Users/Влад/AppData/Local/Programs/Python/Python310/python.exe c:/python_projects/mirea_sem3_tfl_1.py
Введите выражение: (1 + 2) * 2
Выражение в обратной польской записи: 12+2*
PS C:\Users\Влад>

Рисунок 1 – Результат теста №1.1



```
C: > python_projects > mirea_sem3_tfl_1.py > my_stack
1 my_stack, res = list(), str() # "стек" и строка для формирования результата
2 expression = input('Введите выражение: ') # ввод алгебраического выражения
3 symbols = {'*':3, '/':3, '+':2, '-':2, '(':1} # словарь для определения приоритетов операций
4 for i in expression:
5     if i.isdigit() or i.isalpha(): # текущий символ - буква или цифра
6         res+=i # прибавляем к строке результата
7     elif i in '+-/*': # поступил символ операции
8         if not len(my_stack) or symbols[my_stack[-1]]<symbols[i]: # Если стек все еще пуст, или
9             my_stack.append(i) # помещаем текущий символ в стек
10        elif symbols[my_stack[-1]]>=symbols[i]: # символ, находящийся на вершине стека имеет п
11            while len(my_stack) and symbols[my_stack[-1]]>=symbols[i]: # извлекаем символы из
12                res+=my_stack.pop()
13            my_stack.append(i) # добавляем символ в стек
14        elif i=='(': # Если текущий символ - открывающая скобка, то помещаем ее в стек
15            my_stack.append(i)
16        elif i==')': # Если текущий символ - закрывающая скобка, то извлекаем символы из стека в
17            while my_stack[-1]!='(':
18                res+=my_stack.pop()
19            del my_stack[-1]
20 print(f"Выражение в обратной польской записи: {res+''.join(reversed(my_stack))}") # вывод выра
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

Введите выражение: (8+2*5)/(1+3*2-4)
Выражение в обратной польской записи: 825*+132*+4-/
PS C:\Users\Влад>

Рисунок 2 – Результат теста №1.2

```
mirea_sem3_tfl_1.py X
C: > python_projects > mirea_sem3_tfl_1.py > ...
1 my_stack,res = list(),str() #"стек" и строка для формирования результата
2 expression = input('Введите выражение: ') #ввод алгебраического выражения
3 symbols = {'*':3,'/':3,'+':2,'-':2,'(':1} #словарь для определения приоритетов операций
4 for i in expression:
5     if i.isdigit() or i.isalpha(): #текущий символ - буква или цифра
6         res+=i #прибавляем к строке результата
7     elif i in '*/+-': #поступил символ операции
8         if not len(my_stack) or symbols[my_stack[-1]]<symbols[i]: #Если стек все еще пуст, или находящиеся в
9             my_stack.append(i) #помещаем текущий символ в стек
10        elif symbols[my_stack[-1]]>=symbols[i]: #символ, находящийся на вершине стека имеет приоритет, больш
11            while len(my_stack) and symbols[my_stack[-1]]>=symbols[i]: # извлекаем символы из стека в выходн
12                res+=my_stack.pop()
13            my_stack.append(i) #добавляем символ в стек
14        elif i=='(': #Если текущий символ - открывающая скобка, то помещаем ее в стек
15            my_stack.append(i)
16        elif i==')': #Если текущий символ - закрывающая скобка, то извлекаем символы из стека в выходную строку
17            while my_stack[-1]!='(':
18                res+=my_stack.pop()
19            del my_stack[-1]
20 print(f"Выражение в обратной польской записи: {res+''.join(reversed(my_stack))}") #вывод выражения в ОПЗ

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

Введите выражение: 2+2
Выражение в обратной польской записи: 22+
PS C:\Users\Влад>
```

Рисунок 3 – Результат теста №1.3

```
mirea_sem3_tfl_1.py X
C: > python_projects > mirea_sem3_tfl_1.py > ...
1 my_stack,res = list(),str() #"стек" и строка для формирования результата
2 expression = input('Введите выражение: ') #ввод алгебраического выражения
3 symbols = {'*':3,'/':3,'+':2,'-':2,'(':1} #словарь для определения приоритетов операций
4 for i in expression:
5     if i.isdigit() or i.isalpha(): #текущий символ - буква или цифра
6         res+=i #прибавляем к строке результата
7     elif i in '*/+-': #поступил символ операции
8         if not len(my_stack) or symbols[my_stack[-1]]<symbols[i]: #Если стек все еще пуст, или находящиеся в нем символы имеют мене
9             my_stack.append(i) #помещаем текущий символ в стек
10        elif symbols[my_stack[-1]]>=symbols[i]: #символ, находящийся на вершине стека имеет приоритет, больший или равный приоритет
11            while len(my_stack) and symbols[my_stack[-1]]>=symbols[i]: # извлекаем символы из стека в выходную строку до тех пор, п
12                res+=my_stack.pop()
13            my_stack.append(i) #добавляем символ в стек
14        elif i=='(': #Если текущий символ - открывающая скобка, то помещаем ее в стек
15            my_stack.append(i)
16        elif i==')': #Если текущий символ - закрывающая скобка, то извлекаем символы из стека в выходную строку до тех пор, пока не вст
17            while my_stack[-1]!='(':
18                res+=my_stack.pop()
19            del my_stack[-1]
20 print(f"Выражение в обратной польской записи: {res+''.join(reversed(my_stack))}") #вывод выражения в ОПЗ

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

Введите выражение: ((5+7*2-1)/(2*1+1))+1
Выражение в обратной польской записи: 572*+1-21*1+/1+
PS C:\Users\Влад>
```

Рисунок 4 – Результат теста №1.4

2 ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2

2.1 ПОСТАНОВКА ЗАДАЧИ

Разработать программу на выбранном языке программирования для вычисления значения выражения, записанного в обратной польской записи.

2.2 КОД ПРОГРАММЫ

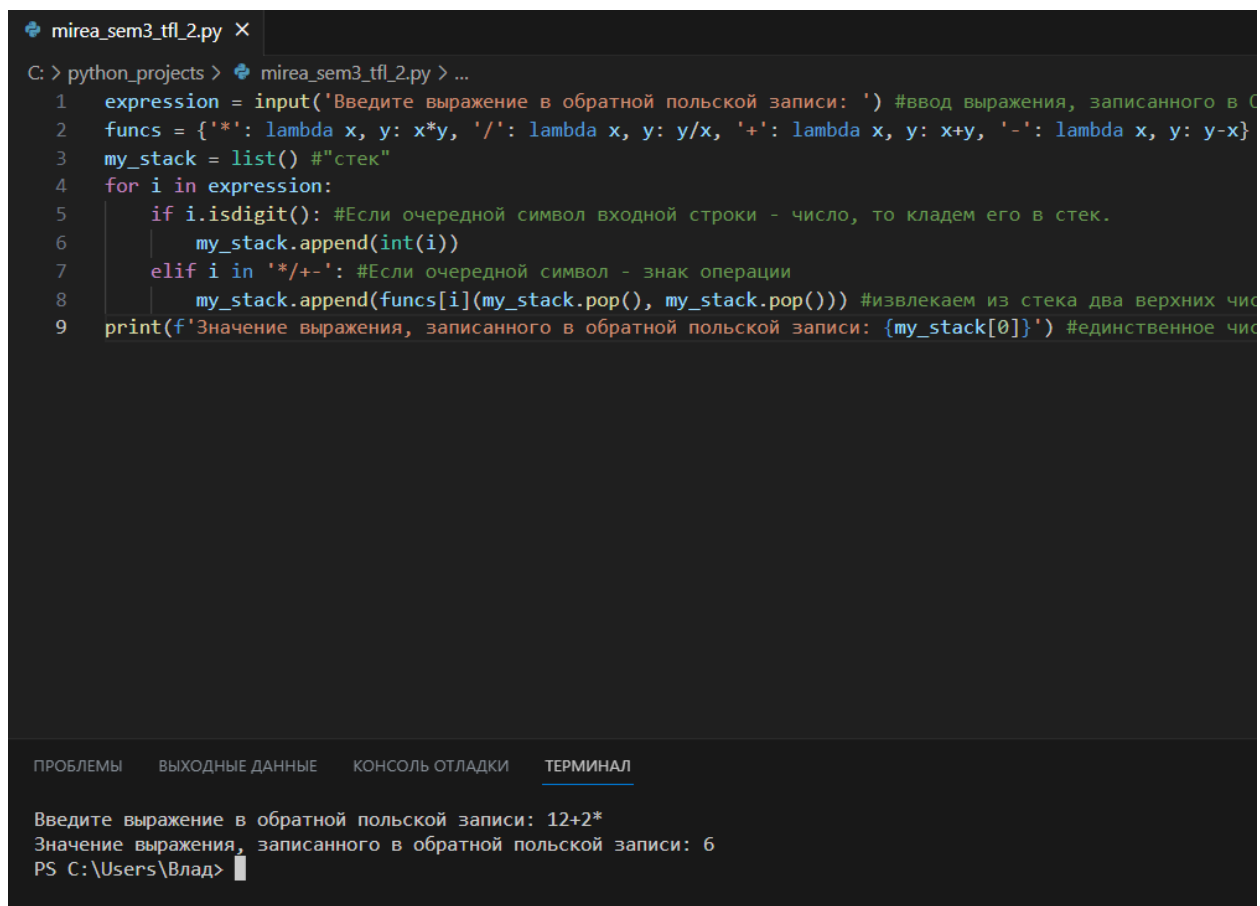
```
expression = input('Введите выражение в обратной польской записи: ') #ввод
выражения, записанного в ОПЗ
funcs = {'*': lambda x, y: x*y, '/': lambda x, y: y/x, '+': lambda x, y: x+y, '-': lambda x, y: y-x} #словарь, содержащий операции в качестве ключей и функции, соответствующие этим операциям
my_stack = list() #"стек"
for i in expression:
    if i.isdigit(): #Если очередной символ входной строки - число, то кладем его в стек.
        my_stack.append(int(i))
    elif i in '*/+-': #Если очередной символ - знак операции
        my_stack.append(funcs[i](my_stack.pop(), my_stack.pop())) #извлекаем из стека два верхних числа, используем их в качестве операндов для этой операции, затем кладем результат обратно в стек
print(f'Значение выражения, записанного в обратной польской записи: {my_stack[0]}') #единственное число в стеке - результат
```

2.3 НАБОР ТЕСТОВ

Таблица 2. Таблица тестирования программы

Вычисление значения выражения, записанного в обратной польской записи		
Номер теста	Входные данные	Эталон результата
1	$12+2^*$	6
2	$825^*+132^*+4- /$	6
3	$22+$	4

2.4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ



```
mirea_sem3_tfl_2.py X
C: > python_projects > mirea_sem3_tfl_2.py > ...
1 expression = input('Введите выражение в обратной польской записи: ') #ввод выражения, записанного в О
2 funcs = {'*': lambda x, y: x*y, '/': lambda x, y: y/x, '+': lambda x, y: x+y, '-': lambda x, y: y-x}
3 my_stack = list() #стек
4 for i in expression:
5     if i.isdigit(): #Если очередной символ входной строки - число, то кладем его в стек.
6         my_stack.append(int(i))
7     elif i in '*/+-': #Если очередной символ - знак операции
8         my_stack.append(funcs[i](my_stack.pop(), my_stack.pop())) #извлекаем из стека два верхних чис
9 print(f'Значение выражения, записанного в обратной польской записи: {my_stack[0]}') #единственное чис

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

Введите выражение в обратной польской записи: 12+2*
Значение выражения, записанного в обратной польской записи: 6
PS C:\Users\Влад>
```

Рисунок 5 – Результат теста №2.1

```
mirea_sem3_tfl_2.py X
C: > python_projects > mirea_sem3_tfl_2.py > ...
1 expression = input('Введите выражение в обратной польской записи: ') #ввод выражения, записанного в ОПЗ
2 funcs = {'*': lambda x, y: x*y, '/': lambda x, y: y/x, '+': lambda x, y: x+y, '-': lambda x, y: x-y} #словарь, со
3 my_stack = list() #стек
4 for i in expression:
5     if i.isdigit(): #Если очередной символ входной строки - число, то кладем его в стек.
6         my_stack.append(int(i))
7     elif i in '*+/-': #Если очередной символ - знак операции
8         my_stack.append(funcs[i](my_stack.pop(), my_stack.pop())) #извлекаем из стека два
9 print(f'Значение выражения, записанного в обратной польской записи: {my_stack[0]}') #единственное число в стеке -

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

PS C:\Users\Влад> & C:/Users/Влад/AppData/Local/Programs/Python/Python310/python.exe c:/python_projects/mirea_sem3_tfl_2.py
Введите выражение в обратной польской записи: 825*+132*+4-/
Значение выражения, записанного в обратной польской записи: 6.0
PS C:\Users\Влад>
```

Рисунок 6 – Результат теста №2.2

```
mirea_sem3_tfl_2.py X
C: > python_projects > mirea_sem3_tfl_2.py > ...
1 expression = input('Введите выражение в обратной польской записи: ') #ввод выражения, записанного в ОПЗ
2 funcs = {'*': lambda x, y: x*y, '/': lambda x, y: y/x, '+': lambda x, y: x+y, '-': lambda x, y: x-y} #словарь, со
3 my_stack = list() #стек
4 for i in expression:
5     if i.isdigit(): #Если очередной символ входной строки - число, то кладем его в стек.
6         my_stack.append(int(i))
7     elif i in '*+/-': #Если очередной символ - знак операции
8         my_stack.append(funcs[i](my_stack.pop(), my_stack.pop())) #извлекаем из стека два верхних числа, используем
9 print(f'Значение выражения, записанного в обратной польской записи: {my_stack[0]}') #единственное число в стеке -

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

PS C:\Users\Влад> & C:/Users/Влад/AppData/Local/Programs/Python/Python310/python.exe c:/python_projects/mirea_sem3_tfl_2.py
Введите выражение в обратной польской записи: 825*+132*+4-/
Значение выражения, записанного в обратной польской записи: 6.0
PS C:\Users\Влад> & C:/Users/Влад/AppData/Local/Programs/Python/Python310/python.exe c:/python_projects/mirea_sem3_tfl_2.py
Введите выражение в обратной польской записи: 22+
Значение выражения, записанного в обратной польской записи: 4
PS C:\Users\Влад>
```

Рисунок 7 – Результат теста №2.3

3 ВЫВОДЫ

В ходе выполнения двух практических работ были получены навыки преобразования арифметического выражения в обратную польскую запись (обратную польскую нотацию), а также навыки подсчёта значения выражения, записанного в обратной польской записи. Для каждой практической работы были разработаны программы на языке Python, реализующие поставленные задачи.

4 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Теория формальных языков. Лекция 1 – СДО (online-edu.mirea.ru)
2. Алгоритмы и методы – обратная польская запись: [Электронный ресурс]. URL: <https://www.interface.ru/home.asp?artid=1492> (дата обращения 04.09.2023)