

Министерство цифрового развития связи и массовых коммуникаций РФ

Государственное бюджетное образовательное учреждение высшего
образования

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра «Математическая кибернетика и информационные технологии»

дисциплина «Структуры и алгоритмы обработки данных»

Отчет по лабораторной работе №4

«Реализация стека/дека»

Подготовил: студент группы

БВТ1903 Саввин Д.И.

Проверил: Кутейников И.А.

Москва

2021

ОГЛАВЛЕНИЕ

1. ЦЕЛЬ РАБОТЫ	3
2. ВЫПОЛНЕНИЕ	5
3. ВЫВОД	20

1. ЦЕЛЬ РАБОТЫ

Разработать программу решающую приведенные ниже задачи с помощью стека/дека.

Задания:

1. Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.

2. Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.

3. Даны три стержня и n дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести n дисков со стержня А на стержень С, сохранив их первоначальный порядок. При переносе дисков необходимо соблюдать следующие правила: - на каждом шаге со стержня на стержень переносить только один диск; - диск нельзя помещать на диск меньшего размера; - для промежуточного хранения можно использовать стержень В. Реализовать алгоритм, используя три стека вместо стержней А, В, С. Информация о дисках хранится в исходном файле.

4. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.

5. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек.

6. Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов.

7. Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.

8. Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

9. Дан текстовый файл. Используя стек, вычислить значение логического выражения, записанного в текстовом файле в следующей форме: $\langle \text{ЛВ} \rangle ::= T \mid F \mid (N) \mid (A) \mid (X) \mid (O)$, где буквами обозначены логические константы и операции: T – True, F – False, N – Not, A – And, X – Xor, O – Or.

10. Дан текстовый файл. В текстовом файле записана формула следующего вида: $::= \mid M(,) \mid N(\text{Формула},) \mid \langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$ где буквами обозначены функции: M – определение максимума, N – определение минимума. Используя стек, вычислить значение заданного выражения.

11. Дан текстовый файл. Используя стек, проверить, является ли содержимое текстового файла правильной записью формулы вида: $\langle \text{Формула} \rangle ::= \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle + \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle - \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle \mid \langle \text{Имя} \rangle \mid (\langle \text{Формула} \rangle) \mid \langle \text{Имя} \rangle ::= x \mid y \mid z$

2. ВЫПОЛНЕНИЕ

Ниже представлен код программы test.py, содержащий все методы для решения всех заданий, а также возможностью выбора задания. Названия функций: run_task1(), run_task2(), и т.д.

test.py

```
import deque
import stack

def name_count(name):
    return ord(name[0])*100 + ord(name[3])*10 + ord(name[6])

# Задание 1
def run_task1():
    str_list = open('in.txt', encoding='utf8').readlines()

    for name in str_list:
        str_list[str_list.index(name)] = str(name).replace('\n', '')

    deq1 = deque.deque([])
    deq2 = deque.deque([])

    sr = int

    # Распределение всех имен по двум декам
    for name in str_list:
        if deq1.is_empty() and deq2.is_empty():
            deq1.add_start(name)
            sr = name_count(name)
        else:
            last = deq1.remove_end()
            deq1.add_end(last)
            if name_count(name) < sr:
                deq1.add_start(name)
            elif name_count(name) > name_count(last):
                deq1.add_end(name)
            else:
                if deq2.is_empty():
                    deq2.add_start(name)
                    sr2 = name_count(name)
                else:
                    last2 = deq2.remove_end()
                    deq2.add_end(last2)
                    if name_count(name) < sr2:
                        deq2.add_start(name)
                    else:
                        deq2.add_end(name)

    if not deq1.is_empty():
        sr = deq1.remove_start()
        deq1.add_start(sr)
        sr = name_count(sr)
    if not deq2.is_empty():
        sr2 = deq2.remove_start()
        deq2.add_start(sr2)
        sr2 = name_count(sr2)
```

```

# Слияние двух деков
in_memory = []
first_shag = True
while not deq2.is_empty():
    if first_shag:
        name = deq2.remove_start()
        first_shag = False
    while not first_shag:
        first = deq1.remove_start()
        deq1.add_start(first)
        if name_count(name) < name_count(first):
            deq1.add_start(name)
            deq1.deque = list(in_memory) + list(deq1.deque)
            first_shag = True
            in_memory = []
        else:
            in_memory.append(deq1.remove_start())

f = open('out.txt', 'w', encoding='utf8')
for i in deq1.deque:
    f.write(i + '\n')

# Задание 2
def run_task2():

    deq = deque.deque(['r', 't', 's', 'y', 'p', 'o', 'x', 'f', 'm', 'b', 'q',
                       'd', 'a', 'g', 'n', 'v', 'u', 'i', 'k', 'h', 'j', 'l',
                       'z', 'c', 'e', 'w'])

    text = open('in2.txt', 'r', encoding='utf8').read()
    deq.add_start(deq.remove_end())
    translate = ''
    for a in text:
        if not (a == ' '):
            last = deq.remove_end()
            deq.add_end(last)
            if not (last == a):
                while not (last == a):
                    move = deq.remove_end()
                    deq.add_start(move)
                    last = move
            else:
                move = deq.remove_end()
                deq.add_start(move)
                last = move
            if (last == a):
                move = deq.remove_end()
                deq.add_start(move)
                move = deq.remove_end()
                deq.add_start(move)
                translate += str(move)
            else:
                translate += ' '

    open('out2.txt', 'w').write(translate)

# Задание 3
def run_task3():
    def hanoi(n, pin1, pin2, list_of_stacks):
        if n == 1:

```

```

        print(f"Переложить диск 1 со стержня {pin1} на {pin2}")
        list_of_stacks[pin2-1].add(list_of_stacks[pin1-1].remove())
    else:
        hanoi(n - 1, pin1, 6 - pin1 - pin2, list_of_stacks)
        print(f"Переложить диск {n} со стержня {pin1} на {pin2}")
        list_of_stacks[pin2-1].add(list_of_stacks[pin1-1].remove())
        hanoi(n - 1, 6 - pin1 - pin2, pin2, list_of_stacks)

list_of_stacks = [stack.stack([]), stack.stack([]), stack.stack([])]
n = int(input("Введите кол-во дисков\n"))
for i in range(n, 0, -1):
    list_of_stacks[0].add(i)
hanoi(n, 1, 2, list_of_stacks)
print('')

# Задание 4
def run_task4():
    text = open('in4.txt', 'r', encoding='utf8').read()
    stk = stack.stack([])
    for i in range(0, 1000):
        stk.add(1)
    for a in text:
        if a == '(':
            stk.add(1)
        if a == ')':
            stk.remove()
    if len(stk.stack) == 1000:
        print('Баланс скобок соблюден')
    else:
        print('Баланс скобок не соблюден')

# Задание 5
def run_task5():
    text = open('in5.txt', 'r', encoding='utf8').read()
    dq = deque.deque([])
    for i in range(0, 1000):
        dq.add_start(1)
    for a in text:
        if a == '[':
            dq.add_start(1)
        if a == ']':
            dq.remove_start()
    if len(dq.deque) == 1000:
        print('Баланс скобок соблюден')
    else:
        print('Баланс скобок не соблюден')

# Задание 6
def run_task6():
    stck = stack.stack([])
    syms = stack.stack([])
    cyfs = stack.stack([])
    text = open('in6.txt', 'r', encoding='utf8').read()

    for a in text:
        if (ord(a.lower()) > 1071) and (ord(a.lower()) < 1104):
            syms.add(a)
        elif (ord(a) > 47) and (ord(a) < 58):
            cyfs.add(a)
        else:
            stck.add(a)

```

```

cyfs2 = stack.stack([])
syms2 = stack.stack([])
stck2 = stack.stack([])
for i in range(0, len(cyfs.stack)):
    cyfs2.add(cyfs.remove())
for i in range(0, len(syms.stack)):
    syms2.add(syms.remove())
for i in range(0, len(stck.stack)):
    stck2.add(stck.remove())

print(cyfs2.stack)
print(syms2.stack)
print(stck2.stack)

# Задание 7
def run_task7():
    text = open('in7.txt', 'r').read()
    deq = deque.deque([])
    temp_s = ''
    for a in text:
        if a == '-':
            temp_s += a
        if (ord(a) > 47) and (ord(a) < 58):
            temp_s += a
        if a == ' ':
            if int(temp_s) < 0:
                deq.add_end(int(temp_s))
            else:
                deq.add_start(int(temp_s))
            temp_s = ''
    if int(temp_s) < 0:
        deq.add_end(int(temp_s))
    else:
        deq.add_start(int(temp_s))

    otr_deq = deque.deque([])
    a = deq.remove_end()
    while a < 0:
        otr_deq.add_start(a)
        if len(deq.deque) > 0:
            a = deq.remove_end()
    pol_deq = deque.deque([])
    while a > -1:
        pol_deq.add_end(a)
        if len(deq.deque) > 0:
            a = deq.remove_end()
        else:
            a = -2

    print(otr_deq.deque)
    print(pol_deq.deque)

# Задание 8
def run_task8():
    text = open('in8.txt', 'r', encoding='utf8').readlines()

    stck = stack.stack([])
    for a in text:
        stck.add(a)

    pr_s = ''

```



```

for i in range(0, len(stck.stack)):
    s = str(stck.remove())
    if s.__contains__('\n'):
        pr_s += s
    else:
        pr_s += s + '\n'
print(pr_s)

# Задание 9
def run_task9():
    stack1 = stack.stack([])
    stack2 = stack.stack([])

    text = open('in9.txt', 'r', encoding='utf8').read()
    for a in text:
        if a == ')':
            z = stack1.remove()
            x = stack1.remove()
            c = stack2.remove()
            while not ((c == 'A') or (c == 'O') or (c == 'X') or c == 'N'):
                c = stack2.remove()

            if c == 'A':
                stack1.add(z and x)
            if c == 'O':
                stack1.add(z or x)
            if c == 'X':
                if z == x:
                    stack1.add(False)
                else:
                    stack1.add(True)
            if c == 'N':
                stack1.add(x)
                stack1.add(not z)
        else:
            if a == 'T':
                stack1.add(True)
            elif a == 'F':
                stack1.add(False)
            else:
                stack2.add(a)

    print(stack1.remove())

# Задание 10
def run_task10():
    def min(a, b):
        if a <= b:
            return a
        else:
            return b

    def max(a, b):
        if a > b:
            return a
        else:
            return b

    stack1 = stack.stack([])
    stack2 = stack.stack([])
    text = open('in10.txt', 'r', encoding='utf8').read()
    sbros = False

```

```

for a in text:

    if a == ')':
        a = stack1.remove()
        b = stack1.remove()
        c = int
        z = stack2.remove()
        while not ((z == 'M') or (z == 'N')):
            z = stack2.remove()
        if z == 'M':
            stack1.add(max(int(a), int(b)))
        elif z == 'N':
            stack1.add(min(int(a), int(b)))
        sbros = False

    else:
        if (ord(a) > 47) and (ord(a) < 58) and (sbros == False):
            stack1.add(a)
            sbros = True
        elif (ord(a) > 47) and (ord(a) < 58) and (sbros == True):
            temp = stack1.remove() + a
            stack1.add(temp)
        else:
            stack2.add(a)
            sbros = False
print(str(stack1.remove()))

# Задание 11
def run_task11():
    stack1 = stack.stack([])
    stack2 = stack.stack([])
    text = open('in11.txt', 'r', encoding='utf8').read()
    for a in text:
        if (a == 'x') or (a == 'y') or (a == 'z'):
            stack1.add(a)
        else:
            stack2.add(a)

    count_of_xs = 0
    temp = stack1.remove()
    b1 = False
    b2 = False
    b3 = False
    while not (temp == -1):
        if temp == 'x':
            b1 = True
            count_of_xs += 1
        if temp == 'y':
            b2 = True
            count_of_xs += 1
        if temp == 'z':
            b3 = True
            count_of_xs += 1
        temp = stack1.remove()

    bool1 = b1 and b2 and b3
    ch1 = 0
    ch2 = 0
    ch3 = 0
    temp = stack2.remove()
    while not (temp == -1):
        if temp == '(':
            ch1 += 1

```

```

        elif temp == ')':
            ch2 +=1
        else:
            ch3 +=1
            temp = stack2.remove()

    if ch1 == ch2:
        bool2 = True
    else:
        bool2 = False

    if (count_of_xs - ch3) == 1:
        bool3 = True
    else:
        bool3 = False

    print((bool1 == True) and (bool2 == True) and (bool3 == True))

if __name__ == '__main__':
    print('Лабораторная работа 4.\nВыберите задание(1-11)...')
    nomer = input()

    if nomer == '1':
        run_task1()
    elif nomer == '2':
        run_task2()
    elif nomer == '3':
        run_task3()
    elif nomer == '4':
        run_task4()
    elif nomer == '5':
        run_task5()
    elif nomer == '6':
        run_task6()
    elif nomer == '7':
        run_task7()
    elif nomer == '8':
        run_task8()
    elif nomer == '9':
        run_task9()
    elif nomer == '10':
        run_task10()
    elif nomer == '11':
        run_task11()

```

На приведенных ниже рисунках представлены результаты выполнения заданий.

Задание 1.

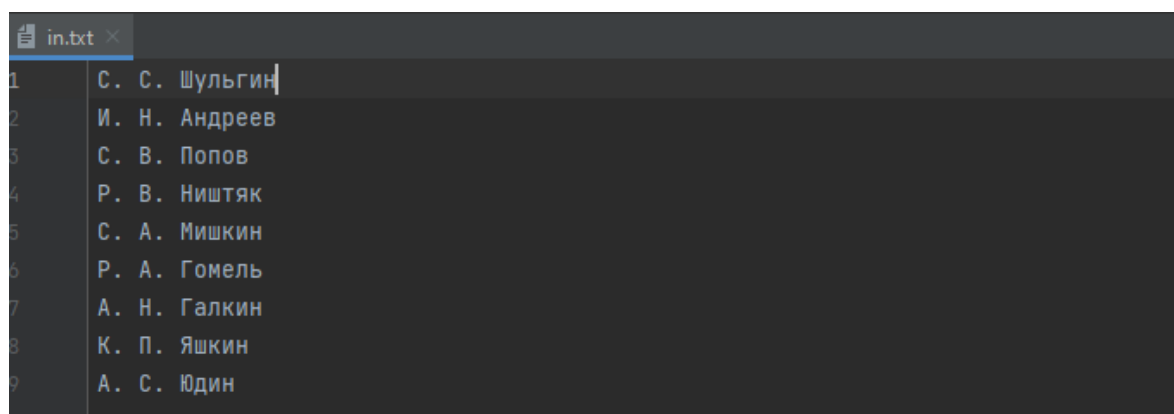


Рис. 1 – Файл ввода in.txt.

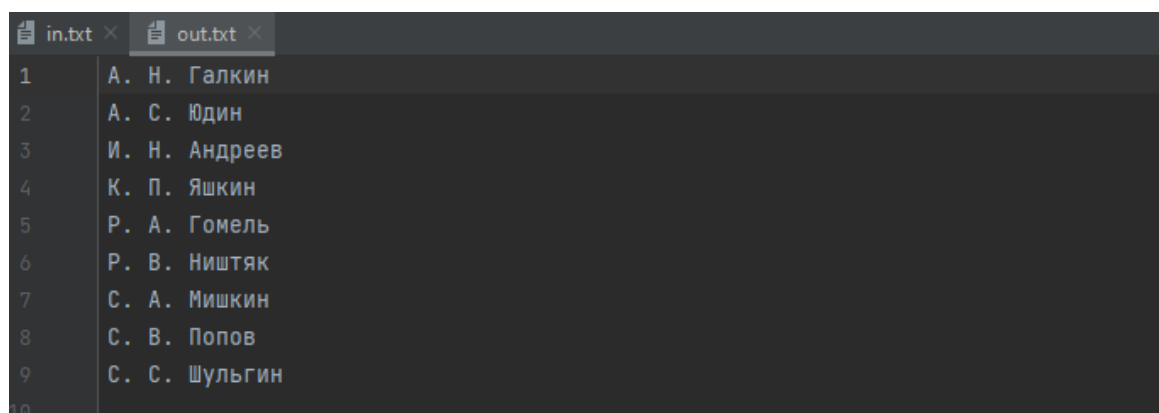


Рис. 2 – Файл вывода out.txt.

Задание 2.

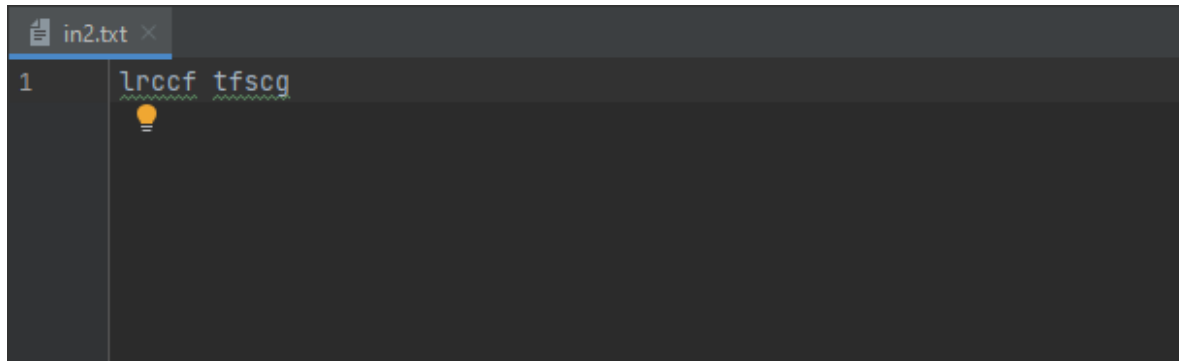


Рис. 3 – Файл ввода in2.txt.

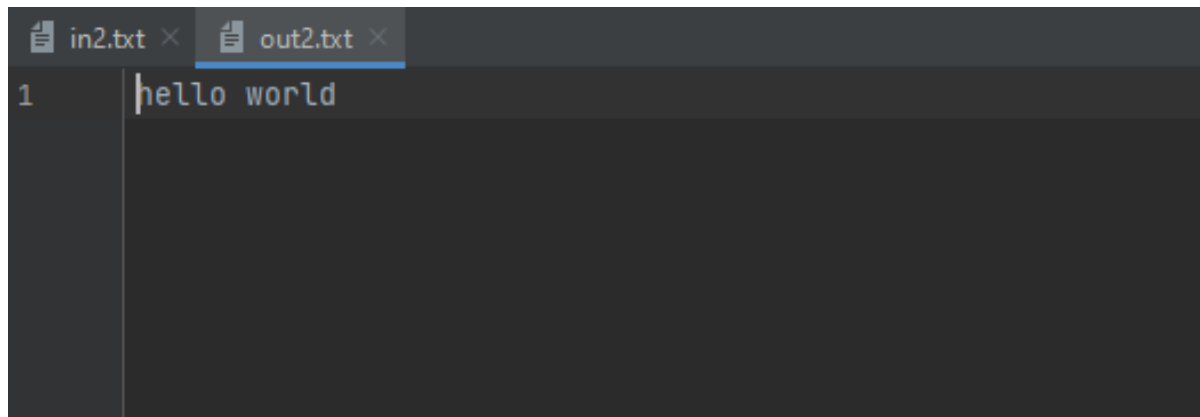


Рис. 4 – Файл вывода out2.txt.

Задание 3.

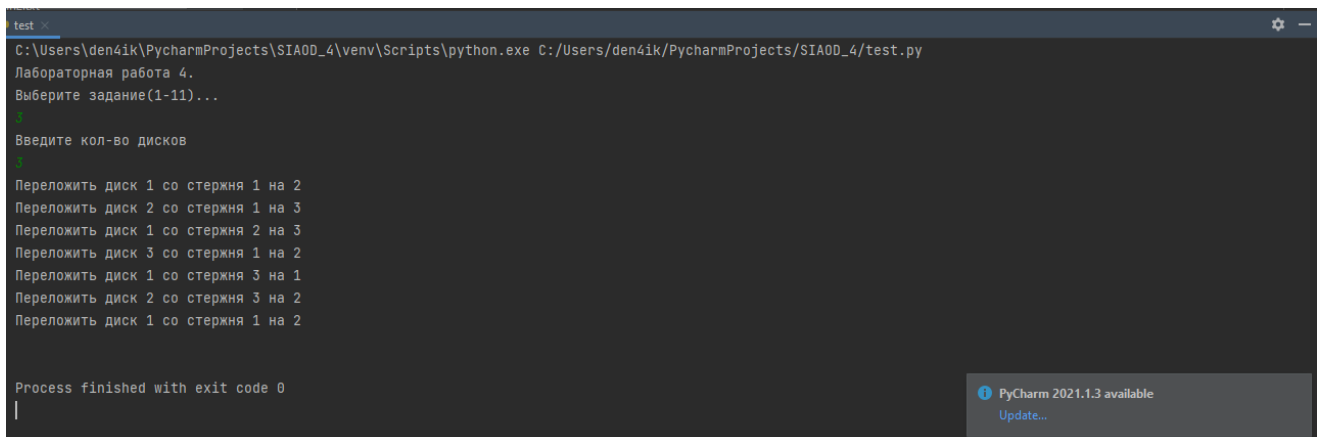


Рис. 5 – Результат работы программы.

Задание 4.

```
in4.txt x
1  алг Треугольник(арг вещ a,b,c, рез лог Otvet)
2    дано | a>0, b>0, c>0, a+b>c, a+c>b, b+c>a
3    надо | Otvet = да, если треугольник равнобедренный
4          | Otvet = нет, если треугольник не равнобедренный
5  нач
6    если (a=b) или (a=c) или (b=c)
7      то Otvet:= да
8      иначе Otvet:= нет
9    все
10  кон
```

Рис. 6 – Файл ввода in4.txt(Вариант 1).

```
in4.txt x
1  алг Треугольник(арг вещ a,b,c, рез лог Otvet)
2    дано | a>0, b>0, c>0, a+b>c, a+c>b, b+c>a
3    надо | Otvet = да, если треугольник равнобедренный
4          | Otvet = нет, если треугольник не равнобедренный
5  нач
6    если (a=b) или (a=c)) или (b=c)
7      то Otvet:= да
8      иначе Otvet:= нет
9    все
10  кон
```

Рис. 7 – Файл ввода in4.txt(Вариант 2).

```
test x
C:\Users\den4ik\PycharmProjects\SIA0D_4\venv\Scripts\python.exe C:/Users/de
Лабораторная работа 4.
Выберите задание(1-11)...
4
Баланс скобок соблюден

Process finished with exit code 0
|
```

Рис. 8 – Результат работы программы(Вариант 1).

```
test x
C:\Users\den4ik\PycharmProjects\SIA0D_4\venv\Scripts\python.exe C:/Users/den4ik
Лабораторная работа 4.
Выберите задание(1-11)...
4
Баланс скобок не соблюден

Process finished with exit code 0
```

Рис. 9 – Результат работы программы(Вариант 2).

Задание 5.

```
in5.txt x
1  алг Треугольник(арг вещь a,b,c, рез лог Otvet)
2    дано | [a>0], [b>0], [c>0], [a+b>c], [a+c>b], [b+c>a]
3    надо | Otvet = да, если треугольник равнобедренный
4          | Otvet = нет, если треугольник не равнобедренный
5    нач
6      если (a=b) или (a=c) или (b=c)
7      то Otvet:= да
8      иначе Otvet:= нет
9    все
10   кон
```

Рис. 10 – Файл ввода in5.txt(Вариант 1).

```
in5.txt x
1  алг Треугольник(арг вещь a,b,c, рез лог Otvet)
2    дано | [a>0, [b>0], [c>0], [a+b>c], [a+c>b], [b+c>a]
3    надо | Otvet = да, если треугольник равнобедренный
4          | Otvet = нет, если треугольник не равнобедренный
5    нач
6      если (a=b) или (a=c) или (b=c)
7      то Otvet:= да
8      иначе Otvet:= нет
9    все
10   кон
```

Рис. 11 – Файл ввода in5.txt(Вариант 2).

```
test x
C:\Users\den4ik\PycharmProjects\SIA00_4\venv\Scripts\python.exe C:/Users/den4ik/
Лабораторная работа 4.
Выберите задание(1-11)...
5
Баланс скобок соблюден

Process finished with exit code 0
|
```

Рис. 12 – Результат работы программы(Вариант 1).

```
test x
C:\Users\den4ik\PycharmProjects\SIA00_4\venv\Scripts\python.exe C:/Users/
Лабораторная работа 4.
Выберите задание(1-11)...
5
Баланс скобок не соблюден

Process finished with exit code 0
```

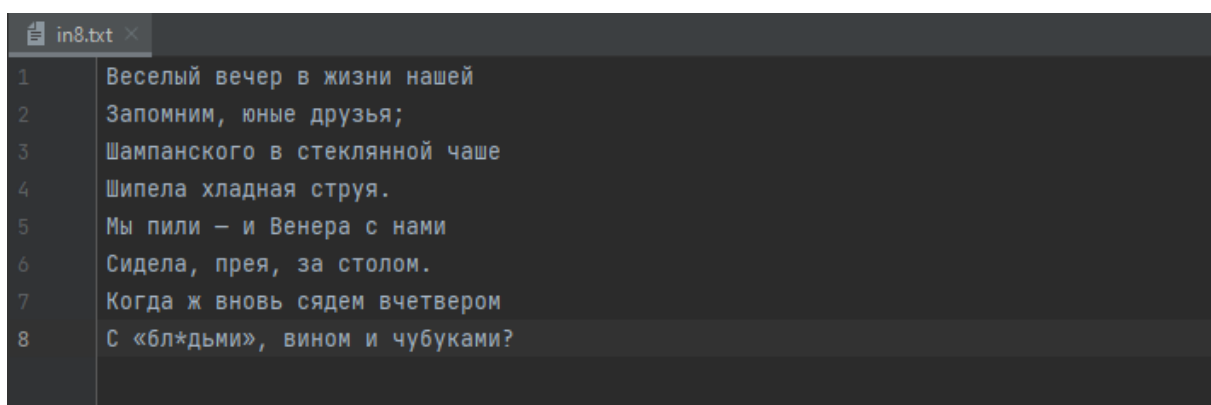
Рис. 13 – Результат работы программы(Вариант 2).

Задание 6.

Вместе с этим возросли и военные расходы: с 1 млрд долларов в 1933 году до 6,5 млрд в 1938 году. Если в 1933 году военные третьего рейха составляли 4 % от всего бюджета, в 1934 - 18 %, то в 1936 – уже 39 %. А в 1938 году на них приходилось 50 %. При помощи компаний США, Великобритании и Франции в короткий срок в Германии было построено более 300 крупных военных заводов. ✓

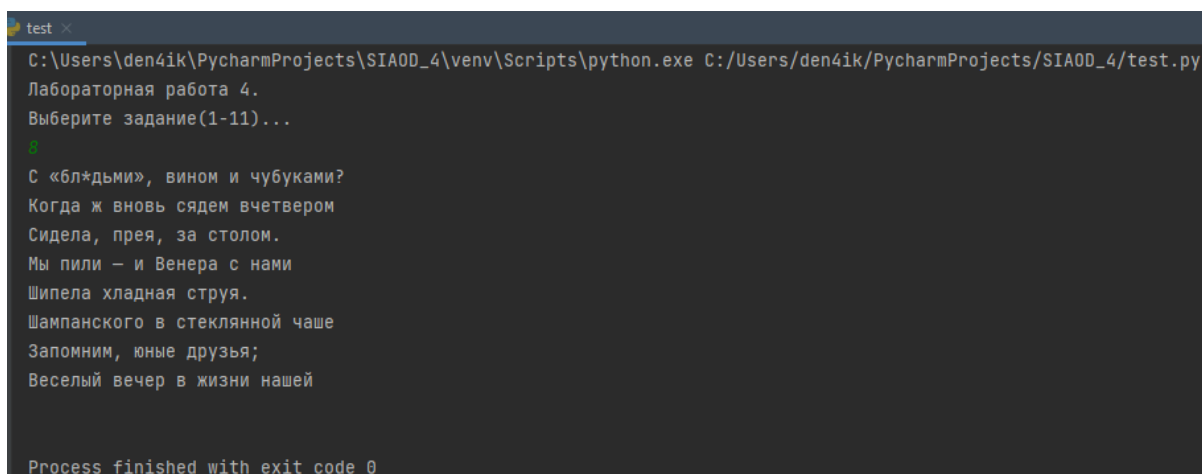
Рис. 14 – Файл ввода in6.txt.

Задание 8.



```
in8.txt ×
1  Веселый вечер в жизни нашей
2  Запомним, юные друзья;
3  Шампанского в стеклянной чаше
4  Шипела холодная струя.
5  Мы пили – и Венера с нами
6  Сидела, преля, за столом.
7  Когда ж вновь сядем вчетвером
8  С «бл*дьми», вином и чубуками?
```

Рис. 18 – Файл ввода in8.txt.

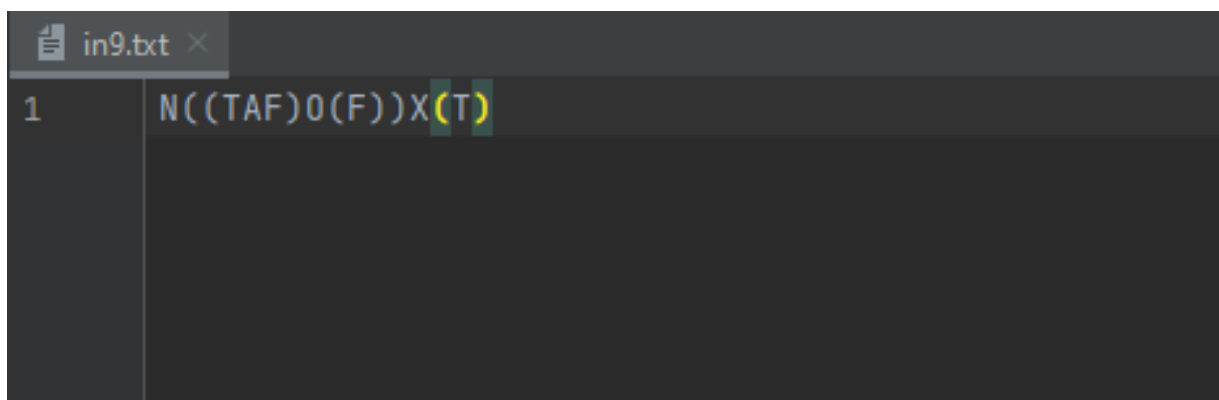


```
test ×
C:\Users\den4ik\PycharmProjects\SIA0D_4\venv\Scripts\python.exe C:/Users/den4ik/PycharmProjects/SIA0D_4/test.py
Лабораторная работа 4.
Выберите задание(1-11)...
8
С «бл*дьми», вином и чубуками?
Когда ж вновь сядем вчетвером
Сидела, преля, за столом.
Мы пили – и Венера с нами
Шипела холодная струя.
Шампанского в стеклянной чаше
Запомним, юные друзья;
Веселый вечер в жизни нашей

Process finished with exit code 0
```

Рис. 19 – Результат работы программы.

Задание 9.



```
in9.txt ×
1  N((TAF)O(F))X(T)
```

Рис. 20 – Файл ввода in9.txt.

```
test x
C:\Users\den4ik\PycharmProjects\SIA0D_4\venv\Scripts\python.exe C:/Users/den4ik/Pychar
Лабораторная работа 4.
Выберите задание(1-11)...
9
False

Process finished with exit code 0
|
```

Рис. 21 – Результат работы программы.

Задание 10.

```
in10.txt x
1 M(N(3,9),M(2,12))
```

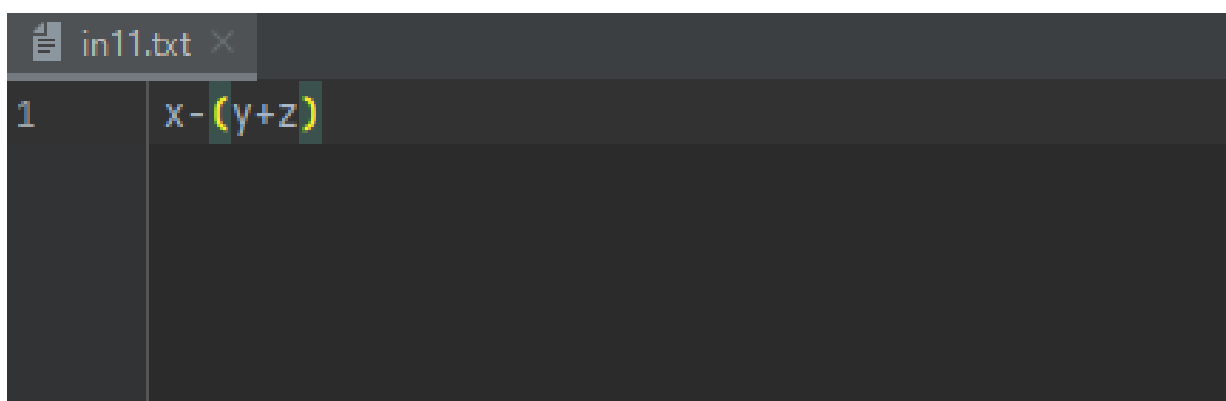
Рис. 22 – Файл ввода in10.txt.

```
test x
C:\Users\den4ik\PycharmProjects\SIA0D_4\venv\Scripts\python.exe C:/U
Лабораторная работа 4.
Выберите задание(1-11)...
10
12

Process finished with exit code 0
```

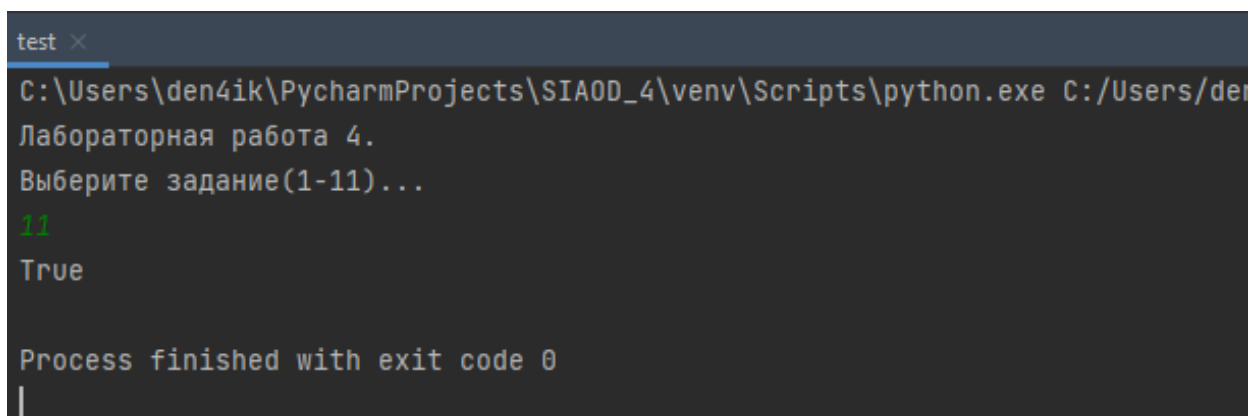
Рис. 23 – Результат работы программы.

Задание 11.



The image shows a text editor window with a single tab labeled 'in11.txt'. The editor contains one line of code on line 1: `x - (y+z)`. The parentheses are highlighted in yellow.

Рис. 24 – Файл ввода in11.txt.



The image shows a terminal window with a tab labeled 'test'. The output of a Python script is displayed as follows:

```
C:\Users\den4ik\PycharmProjects\SIA0D_4\venv\Scripts\python.exe C:/Users/den4ik/PycharmProjects/SIA0D_4/main.py
Лабораторная работа 4.
Выберите задание(1-11)...
11
True

Process finished with exit code 0
```

Рис. 25 – Результат работы программы.

3. ВЫВОД

В ходе данной лабораторной работы были получены и отточены навыки работы с деками и стеками на примере приведенных заданий.