



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

Факультет управления и информатики в технологических системах  
Кафедра информационной безопасности  
Направление подготовки (специальность) 10.05.03 Информационная безопасность  
автоматизированных систем

**Отчет**  
**По технологиям и методам программирования.**  
**Практическая работа №7**  
наименование (вид) дисциплины

Выполнил студент гр. УБ-21  
Заварзин В.Д.  
(ф.и.о.)

Проверил:

Доцент каф. ИБ Маслов А.А.\_\_\_\_\_

\_\_\_\_\_  
(оценка)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

## Вариант 10

Формулировки:

1. На отрезке  $[100, N]$  ( $210 < N < 231$ ) найти количество чисел, составленных из цифр  $a, b, c$ .
2. Составить программу, которая изменяет последовательность слов в строке на обратную.

### Задание 1

```
1 def CountChislo(sub_l, n):
2     k = 0
3     sub_l.sort()
4     for i in range(100, n, 1):
5         l = []
6         l = [int(i/100), int(i/10 % 10), int(i % 10)]
7         l.sort()
8         if sub_l == l:
9             print(i) #подсчёт числа по условию
10            k += 1
11    return k
12
13 nn = int(input("Введите число 210-N-231: "))
14
15 print("Введите до 3-х чисел через пробел из которых может состоять число")
16 sub = [int(i) for i in input().split()]
17
18 print(CountChislo(sub, nn))
```

Run: C:\Users\VLado\PycharmProjects\pythonProject4\venv\Scripts\python.exe C:/Users/VLado/PycharmProjects/pythonProject4/main.py

Введите число 210-N-231: 220

Введите до 3-х чисел через пробел из которых может состоять число

1 2 3

123  
132  
213  
3

Process finished with exit code 0

(рис.1)

```
Введите число 210<N<231: 230
Введите до 3-х чисел через пробел из которых может состоять число
1 2 3
123
132
213
3

Process finished with exit code 0
```

```
def CountChislo(sub_l, n):
    k = 0
    sub_l.sort()
    for i in range(100, n, 1):
        l = []
        l = [int(i/100), int(i/10 % 10), int(i % 10)]
        l.sort()
        if sub_l == l:
            print(i) #подходящие числа под условие
            k += 1
    return k

nn = int(input("Введите число 210<N<231: "))

print("Введите до 3-х чисел через пробел из которых может состоять число")
sub = [int(i) for i in input().split()]

print(CountChislo(sub, nn))
```

(рис.2; рис.3)

Код программы:

```
def CountChislo(sub_l, n):
    k = 0
    sub_l.sort()
    for i in range(100, n, 1):
        l = []
        l = [int(i/100), int(i/10 % 10), int(i % 10)]
        l.sort()
        if sub_l == l:
            print(i) #подходящие числа под условие
            k += 1
    return k

nn = int(input("Введите число 210<N<231: "))

print("Введите до 3-х чисел через пробел из которых  
может состоять число")
sub = [int(i) for i in input().split()]

print(CountChislo(sub, nn))
```

Решение:

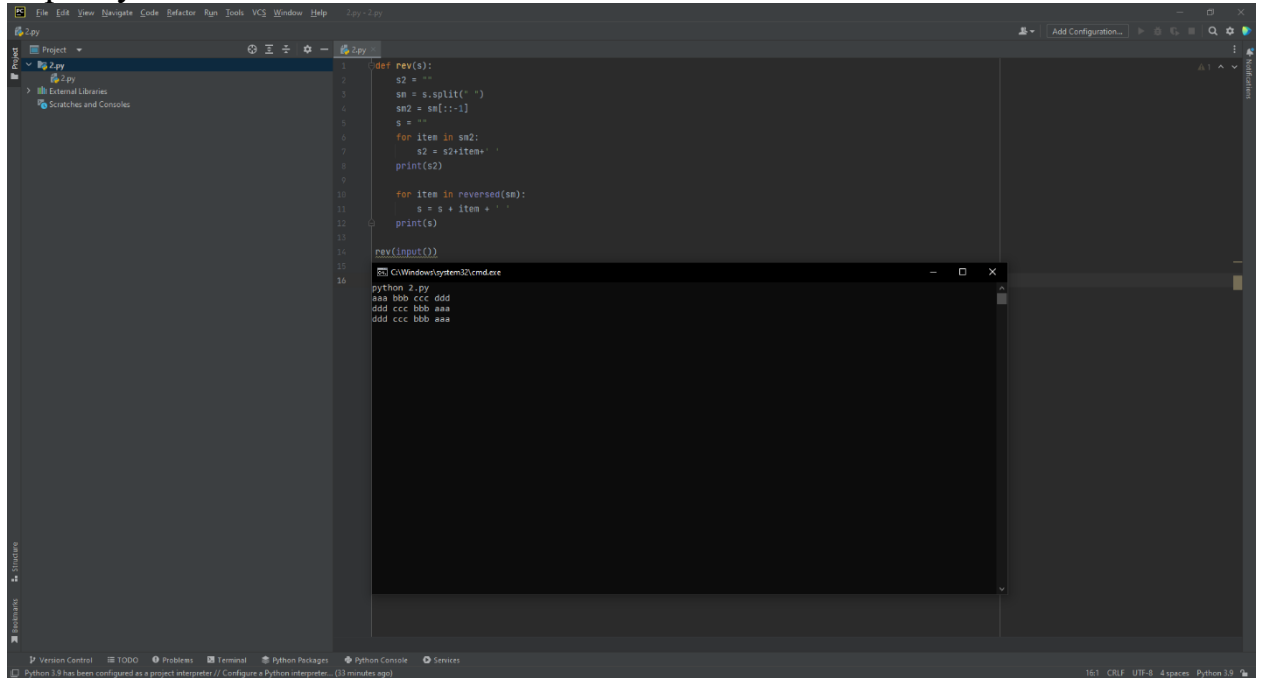
Данная практическая работа заключается в том, чтобы разобраться с работой процедур и функций в *Python*. В этой практической работе мы создаём функцию я назвал её “*CoutChislo*”. Для неё я передаю два значения одно отвечает за цифры, которые должно содержать число и число N, которое нужно по условию.

1. nn, n – число по условию задачи которое вводит пользователь.
2. sub, sub\_l – массив цифр, который должно содержать число.
3. l – массив цифр, который постоянно изменяется в зависимости от перебираемых чисел.
4. k – количество чисел, которые мы получим по условию задачи.
- 5.

Внутри функции мы перебираем числа от 100 до N и сравниваем каждое число с нашими цифрами, в случае совпадения цифр мы выводим число на экран.

## Задание 2

Составить программу, которая изменяет последовательность слов в строке на обратную.



The screenshot shows a Python IDE with a file named `2.py`. The code defines a function `rev(s)` that takes a string `s` and returns a new string with the words in reverse order. The function uses `s.split()` to split the string into words, then iterates over the resulting list in reverse order to build the new string. The main part of the program uses `input()` to get a string from the user and prints the result of `rev(input())`.

```
1 def rev(s):
2     s2 = ""
3     s1 = s.split(" ")
4     s2 = s1[::-1]
5     s = ""
6     for item in s2:
7         s2 = s2+item+ " "
8     print(s2)
9
10    for item in reversed(s1):
11        s = s + item + " "
12    print(s)
13
14
15    rev(input())
16
```

The terminal window shows the execution of the program. The user enters the string `aaa bbb ccc ddd`, and the program outputs `ddd ccc bbb aaa`.

(рис.1)

```

1  def rev(s):
2      s2 = ""
3      sm = s.split(" ")
4      sm2 = sm[::-1]
5      s = ""
6      for item in sm2:
7          s2 = s2+item+' '
8      print(s2)
9
10     for item in reversed(sm):
11         s = s + item + ' '
12     print(s)
13
14     rev(input())

```

```

C:\Windows\system32\cmd.exe

python 2.py
aaa bbb ccc ddd
ddd ccc bbb aaa
ddd ccc bbb aaa

```

(рис.2;рис.3)

Код программы:

```

def rev(s):
    s2 = ""
    sm = s.split(" ")
    sm2 = sm[::-1]
    s = ""
    for item in sm2:
        s2 = s2+item+' '
    print(s2)

    for item in reversed(sm):
        s = s + item + ' '
    print(s)

```

```
rev(input())
```

Решение:

Данная лабораторная работа заключается в том, чтобы разобраться с работой процедур и функций в *Python*. В этой лабораторной работе мы создаём функцию я назвал её “*rev*”. Для неё я передаю строку, которую в последующем мы переворачиваем. В функции представлено сразу два метода изменения последовательности слов в строке на обратную.

1. *s* – исходная и в последующем перевёрнутая строка.
2. *sm* – массив слов в строке.
3. *sm2* – перевёрнутый массив слов.
4. *s2* – перевёрнутая строка.

В первом случае мы разбиваем строку по словам, записываем их в массив и потом при помощи конструкции переворачиваем эти слова в массиве. После чего обратно склеиваем слова в строку. Во втором методе мы в цикле перебираем также массив слов, но мы его сразу передвигаем с конца и также склеиваем в строку.

#### **Вывод:**

Данная лабораторная работа была рассчитана на изучение процедур и функций в *Python*. Выполнив задание я изучил и применил на практике знания по этим подпрограммам.

При выполнении лабораторной работы использовалась среда программирования *PyCharm* (см. рис.1-3). На рис. представлен исходный код программы и пример его работы.