



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 №6

наименование (вид) дисциплины

Выполнил студент гр. УБ-21

Заварзин В.Д.

(ф.и.о.)

Проверил:

Доцент каф. ИБ Маслов А.А.

(оценка)

(подпись)

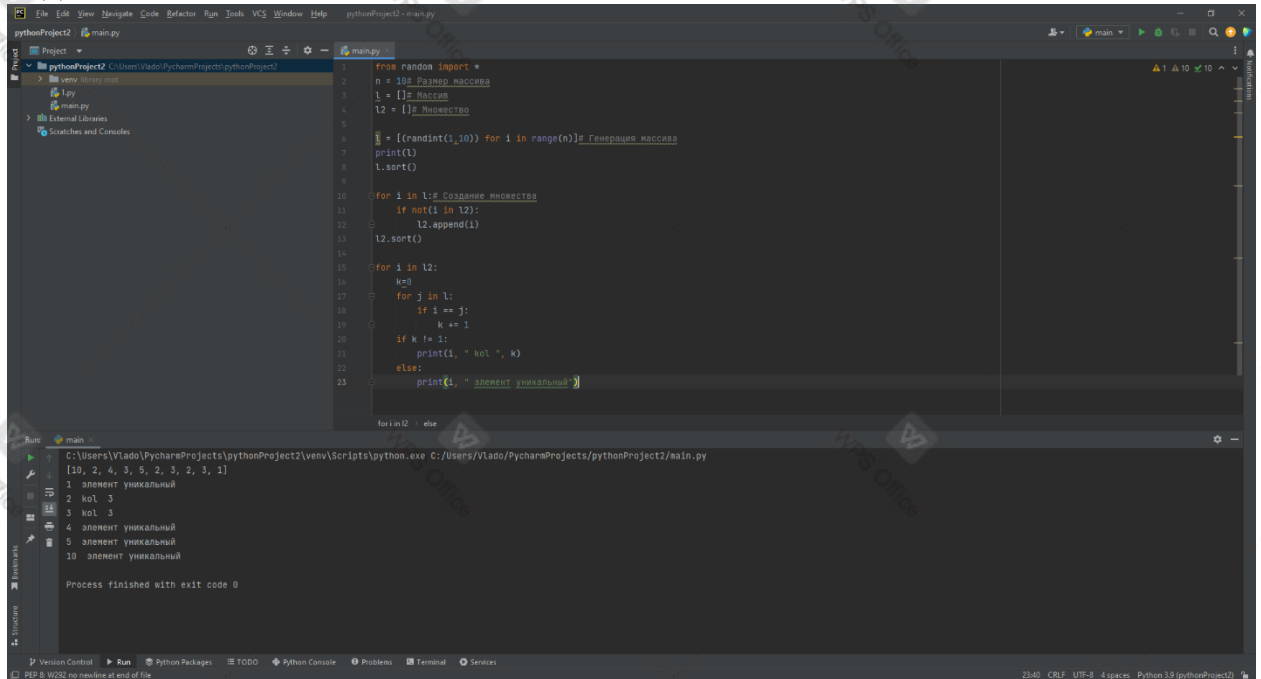
(дата)

Вариант 10

Формулировки:

1. Определите, есть ли в списке повторяющиеся элементы, если да, то вывести на экран это значение, иначе сообщение об их отсутствии.
2. Дан одномерный массив из 15 элементов. Элементам массива меньше 10 присвоить нулевые значения, а элементам больше 20 присвоить 1. Вывести на экран монитора первоначальный и преобразованный массивы в строку.

Задание 1



```
1 from random import *
2 n = 10 # Размер массива
3 l = [] # Массив
4 l2 = [] # Массив
5
6 l = [(randint(1,10)) for i in range(n)] # Генерация массива
7 print(l)
8 l.sort()
9
10 # Создание массива
11 for i in l:
12     if not(i in l2):
13         l2.append(i)
14 l2.sort()
15
16 for i in l2:
17     k = 0
18     for j in l:
19         if i == j:
20             k += 1
21     if k != 1:
22         print(i, " кол. ", k)
23     else:
24         print(i, " элемент уникальный")
25
26 for i in l2:
27     if i < 10:
28         l2[i] = 0
29     elif i > 20:
30         l2[i] = 1
```

Run console output:

```
10, 2, 4, 3, 5, 2, 3, 2, 3, 1)
1 элемент уникальный
2 кол. 3
3 кол. 3
4 элемент уникальный
5 элемент уникальный
10 элемент уникальный
Process finished with exit code 0
```

(рис.1)

```
main.py x
4  l2 = []# Множество
5
6  l = [(randint(1,10)) for i in range(n)]# Генерация массива
7  print(l)
8  l.sort()
9
10 for i in l:# Создание множества
11     if not(i in l2):
12         l2.append(i)
13 l2.sort()
14
15 for i in l2:
16     k=0
17     for j in l:
18         if i == j:
19             k += 1
20     if k != 1:
21         print(i, " kol ", k)
22     else:
23         print(i, " элемент уникальный")

[10, 2, 4, 3, 5, 2, 3, 2, 3, 1]
1 элемент уникальный
2 kol 3
3 kol 3
4 элемент уникальный
5 элемент уникальный
10 элемент уникальный

Process finished with exit code 0
```

(рис.2; рис.3)

Код программы:

```
from random import *
n = 10# Размер массива
l = []# Массив
l2 = []# Множество

l = [(randint(1,10)) for i in range(n)]# Генерация массива
print(l)
l.sort()

for i in l:# Создание множества
    if not(i in l2):
        l2.append(i)
l2.sort()

for i in l2:
    k=0
    for j in l:
        if i == j:
            k += 1
    if k != 1:
        print(i, " kol ", k)
    else:
        print(i, " элемент уникальный")
```

Решение:

Данная лабораторная работа заключается в том, чтобы разобраться с работой одномерных массивов в *Python*. Для начала работы была импортирована библиотека *random*, данная библиотека нам понадобится для того, чтобы быстро генерировать массив для дальнейшей работы.

В этой лабораторной работе нам понадобится 3 ключевые переменные:

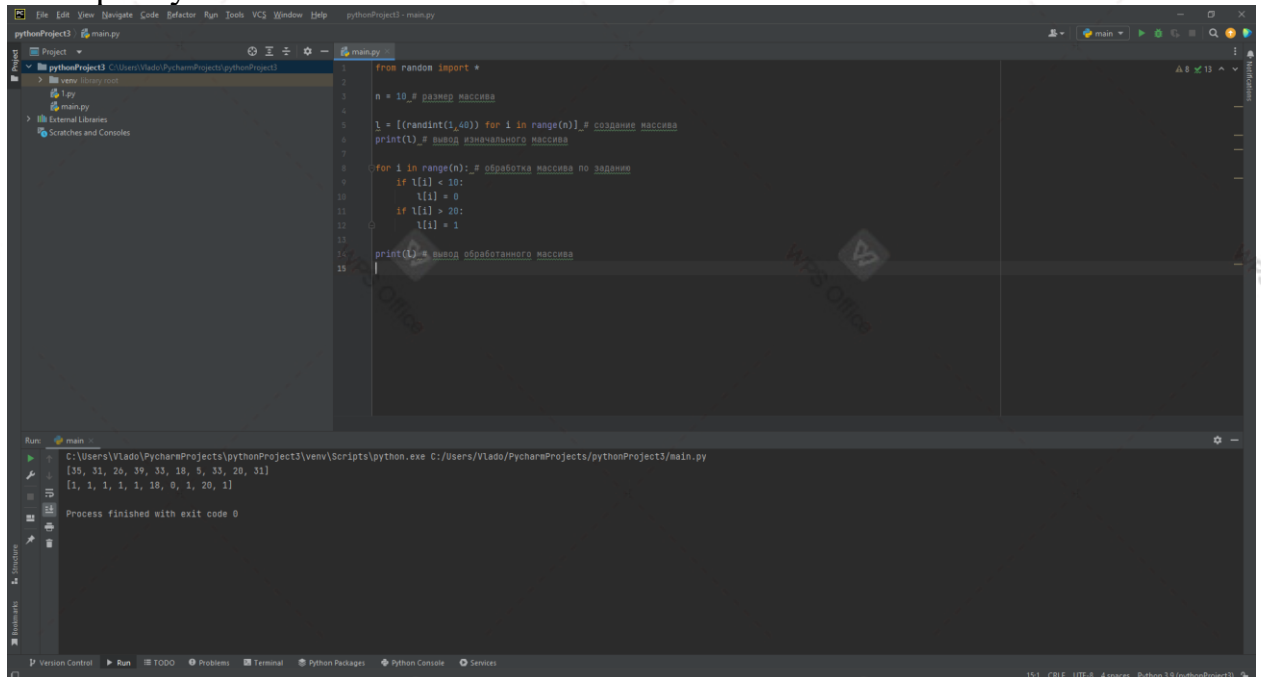
1. *n* – отвечает за размер генерируемого массива
2. *l* – искомый массив
3. *l2* – массив созданный в роли множества

Для начала мы генерируем массив и выводим его для последующей ручной проверки. Далее мы по искомому массиву создаём множество элементов из которых он состоит. После чего мы начинаем перебирать по одному элементу множества, по всему искомому массиву. В случае совпадения мы ведём подсчёт элементов одного типа. После мы выводим. Если при подсчёте элемент оказался всего один, то мы выводим, что элемент является уникальным. Если же он не один, то выводим в формате: ({элемент} kol {количество данного элемента})

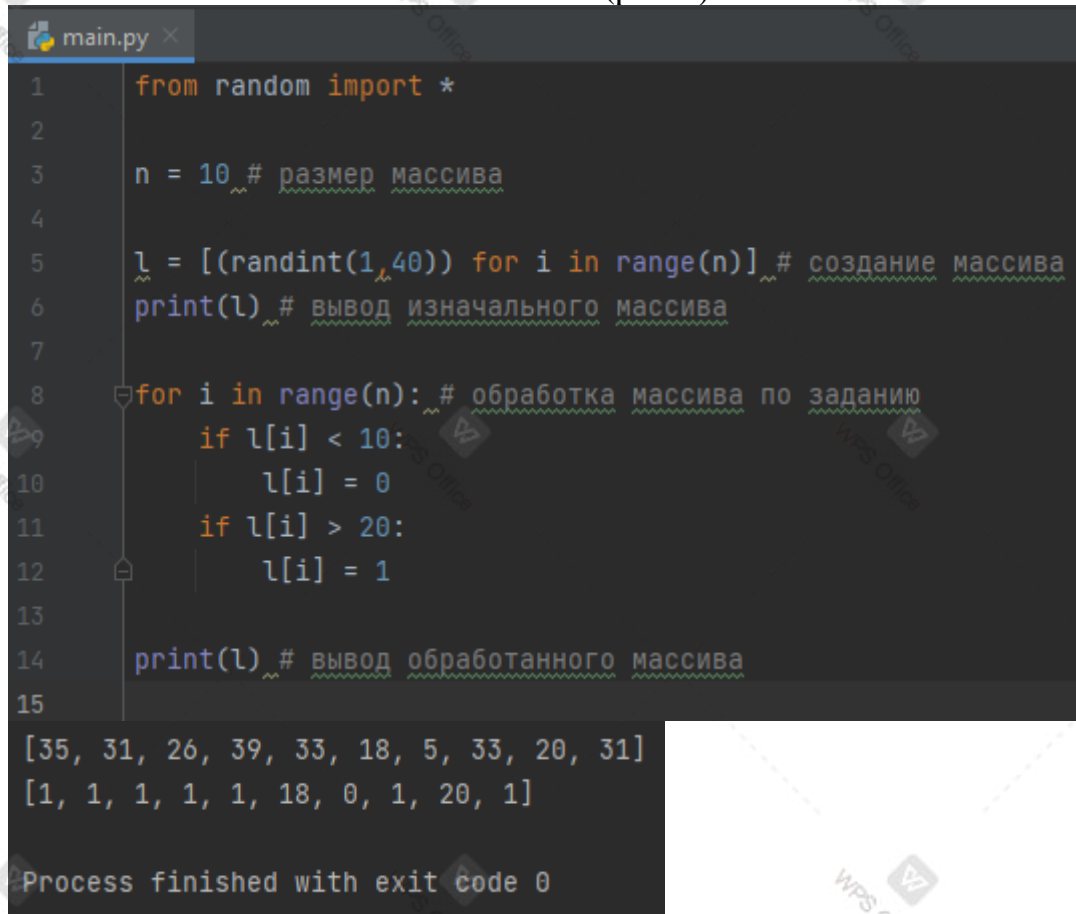
Задание 2

Формулировка:

Дан одномерный массив из 15 элементов. Элементам массива меньше 10 присвоить нулевые значения, а элементам больше 20 присвоить 1. Вывести на экран монитора первоначальный и преобразованный массивы в строчку.



(рис.1)



(рис.2; рис.3)

Код программы:

```
from random import *  
  
n = 10 # размер массива  
  
l = [(randint(1,40)) for i in range(n)] # создание  
массива  
print(l) # вывод изначального массива  
  
for i in range(n): # обработка массива по заданию  
    if l[i] < 10:  
        l[i] = 0  
    if l[i] > 20:  
        l[i] = 1  
  
print(l) # вывод обработанного массива
```

Решение:

Данная лабораторная работа заключается в том, чтобы разобраться с работой одномерных массивов в *Python*. Для начала работы была импортирована библиотека *random*, данная библиотека нам понадобится для того, чтобы быстро генерировать массив для дальнейшей работы.

В этой лабораторной работе нам понадобится 2 ключевые переменные:

1. *n* – отвечает за размер генерируемого массива
2. *l* – искомый массив

Для начала мы генерируем массив и выводим его для последующей ручной проверки. После чего мы начинаем перебирать по одному элементу массива. В случае если встречается элемент массива подходящий под условие форматирования мы изменяем его. Выводим массив полученный после форматирования.

Вывод:

Данная лабораторная работа была рассчитана на изучение одномерных массивов в *Python*. Выполнив задание я изучил и применил на практике знания по одномерным массивам.

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



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)

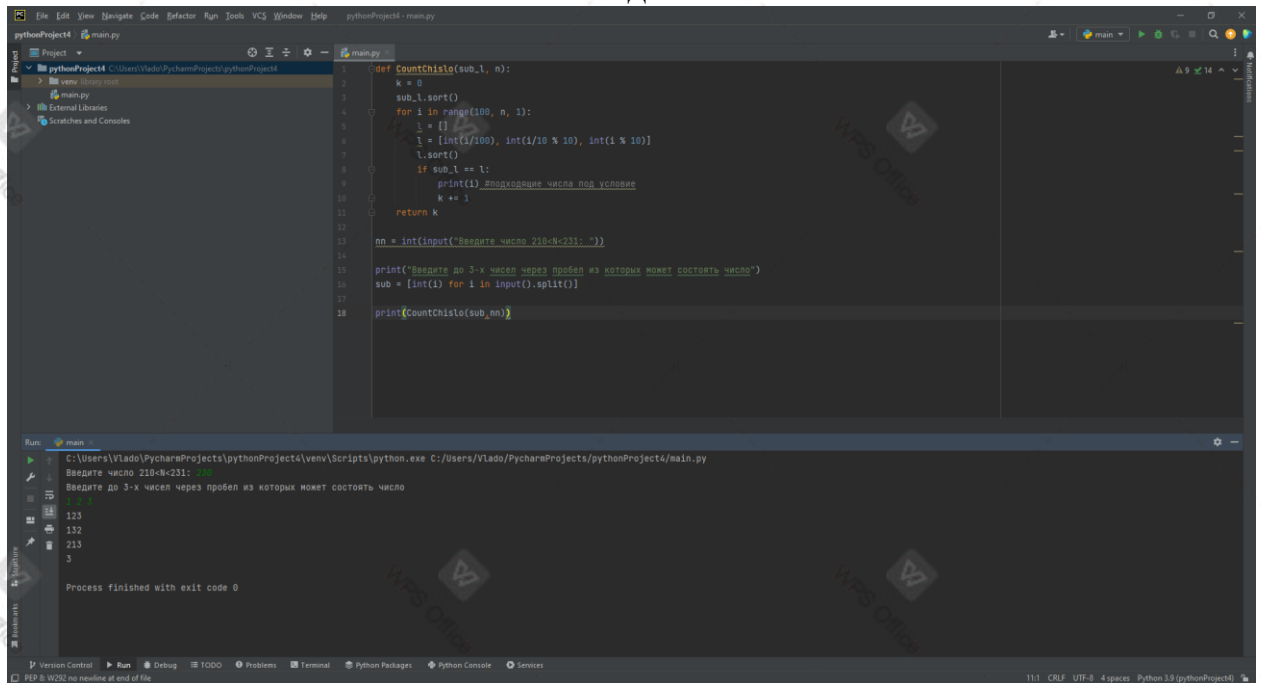
(дата)

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 = [i]
6         l = [int(i/100), int(i/10 % 10), int(i % 10)]
7         l.sort()
8         if sub_l == l:
9             print(i) #выводимое число по условию
10            k += 1
11        return k
12
13 nn = int(input("Введите число 210N-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 210N-231: 210N-231

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_1, n):
    k = 0
    sub_1.sort()
    for i in range(100, n, 1):
        l = []
        l = [int(i/100), int(i/10 % 10), int(i % 10)]
        l.sort()
        if sub_1 == 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:

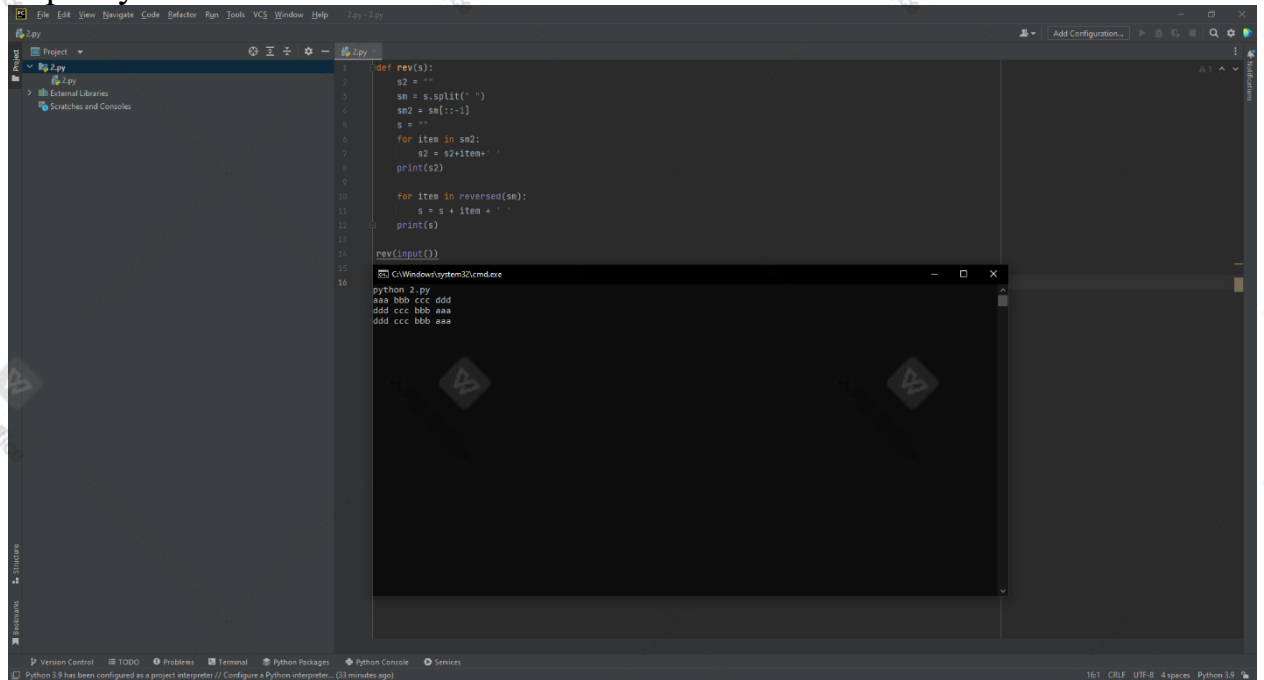
Данная практическая работа заключается в том, чтобы разобраться с работой процедур и функций в *Python*. В этой практической работе мы создаём функцию я назвал её “*CountChislo*”. Для неё я передаю два значения одно отвечает за цифры, которые должно содержать число и число N, которое нужно по условию.

1. nn, n – число по условию задачи которое вводит пользователь.
2. sub, sub_1 – 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 `s2`. The main part of the program prompts the user for input, calls `rev(input())`, and prints the result. A terminal window is open below the code editor, showing the execution of the program with the input `aaa bbb ccc ddd` and the output `ddd ccc bbb aaa`.

```
1 def rev(s):
2     s2 = ""
3     sa = s.split(" ")
4     sa2 = sa[::-1]
5     s = ""
6     for item in sa2:
7         s2 = s2+item+ " "
8     print(s2)
9
10    for item in reversed(sa):
11        s = s + item + " "
12    print(s)
13
14    rev(input())
15
16 python 2.py
aaa bbb ccc ddd
ddd ccc bbb aaa
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.



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 №6

наименование (вид) дисциплины

Выполнил студент гр. УБ-21

Заварзин В.Д.

(ф.и.о.)

Проверил:

Доцент каф. ИБ Маслов А.А.

(оценка)

(подпись)

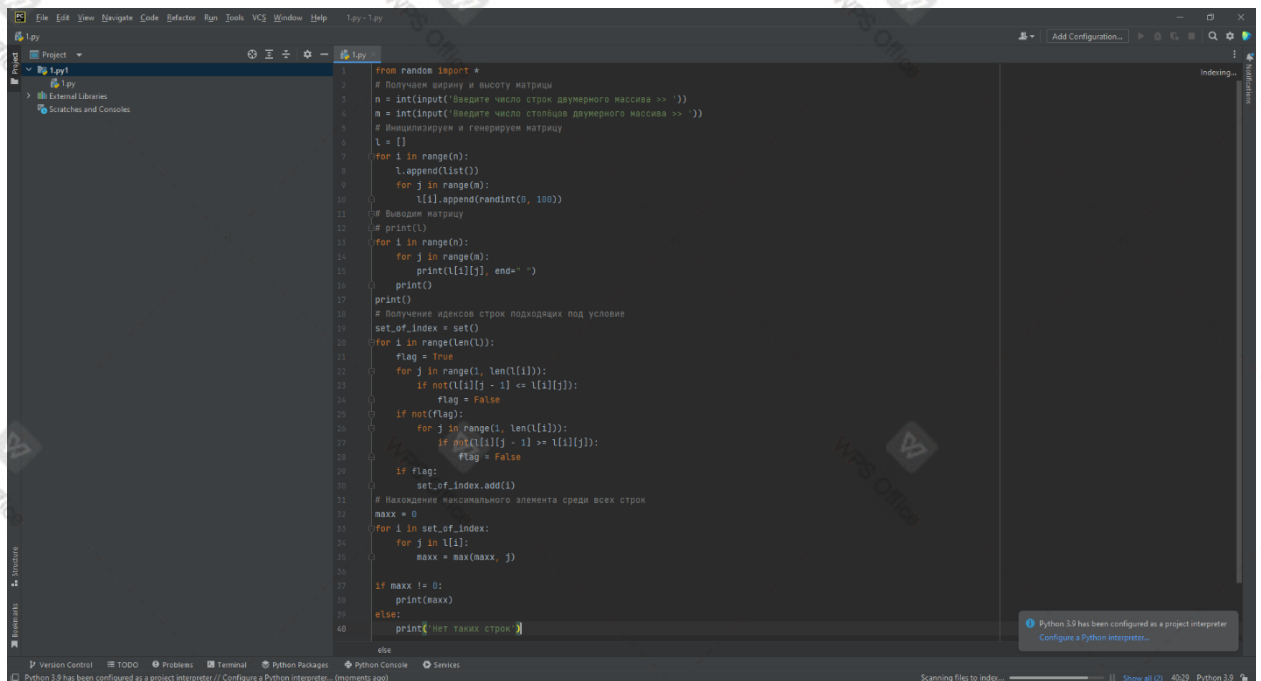
(дата)

Вариант 10

Формулировки:

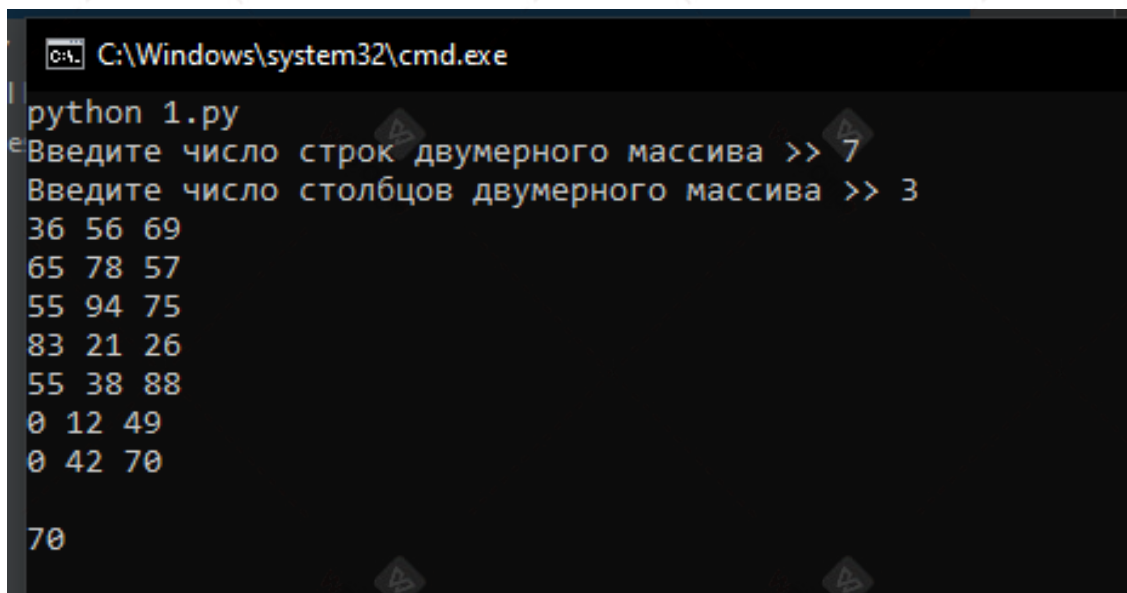
1. Найти максимальный среди всех элементов тех строк заданной матрицы, которые упорядочены (либо по возрастанию, либо по убыванию).
2. Расположить столбцы матрицы $D[M, N]$ в порядке возрастания элементов k -й строки ($1 \leq k \leq M$).

Задание 1



```
1 from random import *
2 # Получаем ширину и высоту матрицы
3 n = int(input("Введите число строк двумерного массива >> "))
4 m = int(input("Введите число столбцов двумерного массива >> "))
5 # Выводим ширину и генерируем матрицу
6 l = []
7 for i in range(n):
8     l.append(list())
9     for j in range(m):
10         l[i].append(randint(0, 100))
11 # Выводим матрицу
12 # print(l)
13 for i in range(n):
14     for j in range(m):
15         print(l[i][j], end=" ")
16     print()
17 print()
18 # Получение индексов строк подходящих под условие
19 set_of_index = set()
20 for i in range(len(l)):
21     flag = True
22     for j in range(1, len(l[i])):
23         if not(l[i][j] - 1 <= l[i][j-1]):
24             flag = False
25     if not(flag):
26         for j in range(1, len(l[i])):
27             if not(l[i][j] - 1 >= l[i][j-1]):
28                 flag = False
29     if flag:
30         set_of_index.add(i)
31 # Нахождение максимального элемента среди всех строк
32 maxx = 0
33 for i in set_of_index:
34     for j in l[i]:
35         maxx = max(maxx, j)
36
37 if maxx != 0:
38     print(maxx)
39 else:
40     print("Нет таких строк")
```

(рис.1)



```
C:\Windows\system32\cmd.exe
python 1.py
Введите число строк двумерного массива >> 7
Введите число столбцов двумерного массива >> 3
36 56 69
65 78 57
55 94 75
83 21 26
55 38 88
0 12 49
0 42 70

70
```

Пример работы программы (рис.2)

```

1  from random import *
2  # Получаем ширину и высоту матрицы
3  n = int(input('Введите число строк двумерного массива >> '))
4  m = int(input('Введите число столбцов двумерного массива >> '))
5  # Инициализируем и генерируем матрицу
6  l = []
7  for i in range(n):
8      l.append(list())
9      for j in range(m):
10         l[i].append(randint(0, 100))
11 # Выводим матрицу
12 # print(l)
13 for i in range(n):
14     for j in range(m):
15         print(l[i][j], end=" ")
16     print()
17 print()
18 # Получение индексов строк подходящих под условие
19 set_of_index = set()
20 for i in range(len(l)):
21     flag = True
22     for j in range(1, len(l[i])):
23         if not(l[i][j - 1] <= l[i][j]):
24             flag = False
25     if not(flag):
26         for j in range(1, len(l[i])):
27             if not(l[i][j - 1] >= l[i][j]):
28                 flag = False
29     if flag:
30         set_of_index.add(i)
31 # Нахождение максимального элемента среди всех строк
32 maxx = 0
33 for i in set_of_index:
34     for j in l[i]:
35         maxx = max(maxx, j)
36
37 if maxx != 0:
38     print(maxx)
39 else:
40     print('Нет таких строк')

```

(рис.3)

Код программы:

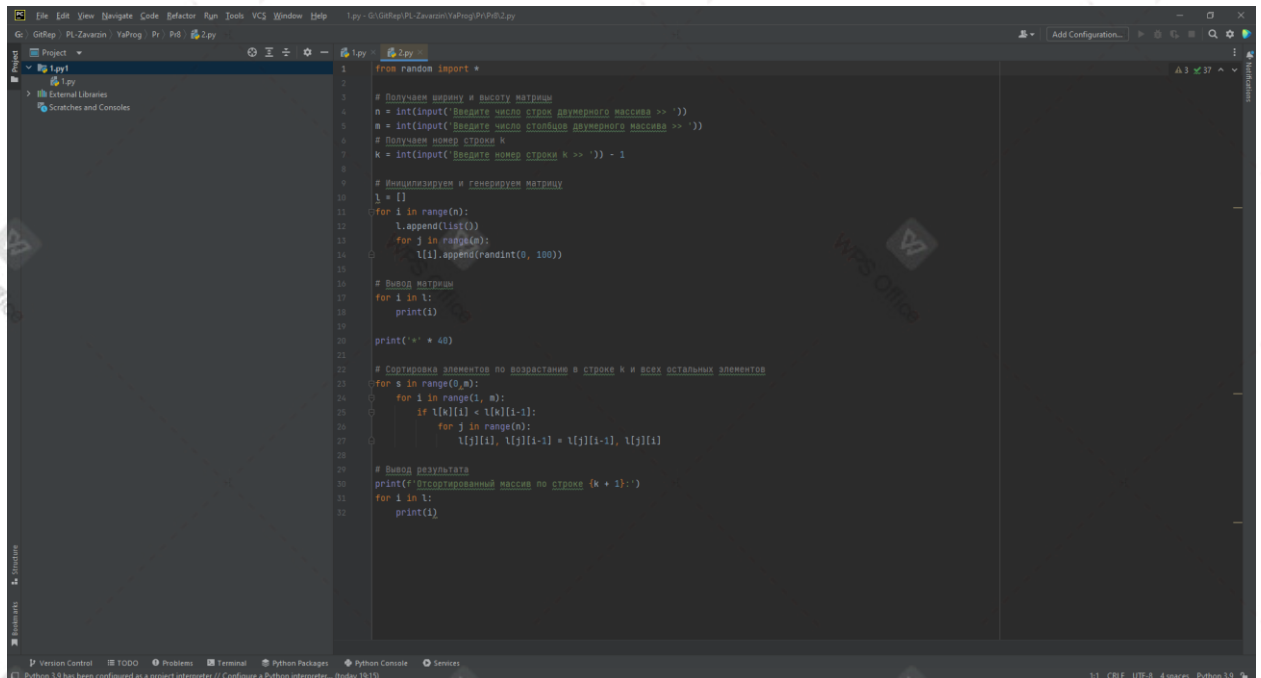
```
from random import *
# Получаем ширину и высоту матрицы
n = int(input('Введите число строк двумерного массива
>> '))
m = int(input('Введите число столбцов двумерного
массива >> '))
# Инициализируем и генерируем матрицу
l = []
for i in range(n):
    l.append(list())
    for j in range(m):
        l[i].append(randint(0, 100))
# Выводим матрицу
# print(l)
for i in range(n):
    for j in range(m):
        print(l[i][j], end=" ")
    print()
print()
# Получение индексов строк подходящих под условие
set_of_index = set()
for i in range(len(l)):
    flag = True
    for j in range(1, len(l[i])):
        if not(l[i][j - 1] <= l[i][j]):
            flag = False
    if not(flag):
        for j in range(1, len(l[i])):
            if not(l[i][j - 1] >= l[i][j]):
                flag = False
    if flag:
        set_of_index.add(i)
# Нахождение максимального элемента среди всех строк
maxx = 0
for i in set_of_index:
    for j in l[i]:
        maxx = max(maxx, j)

if maxx != 0:
    print(maxx)
else:
    print('Нет таких строк')
```

Задание 2

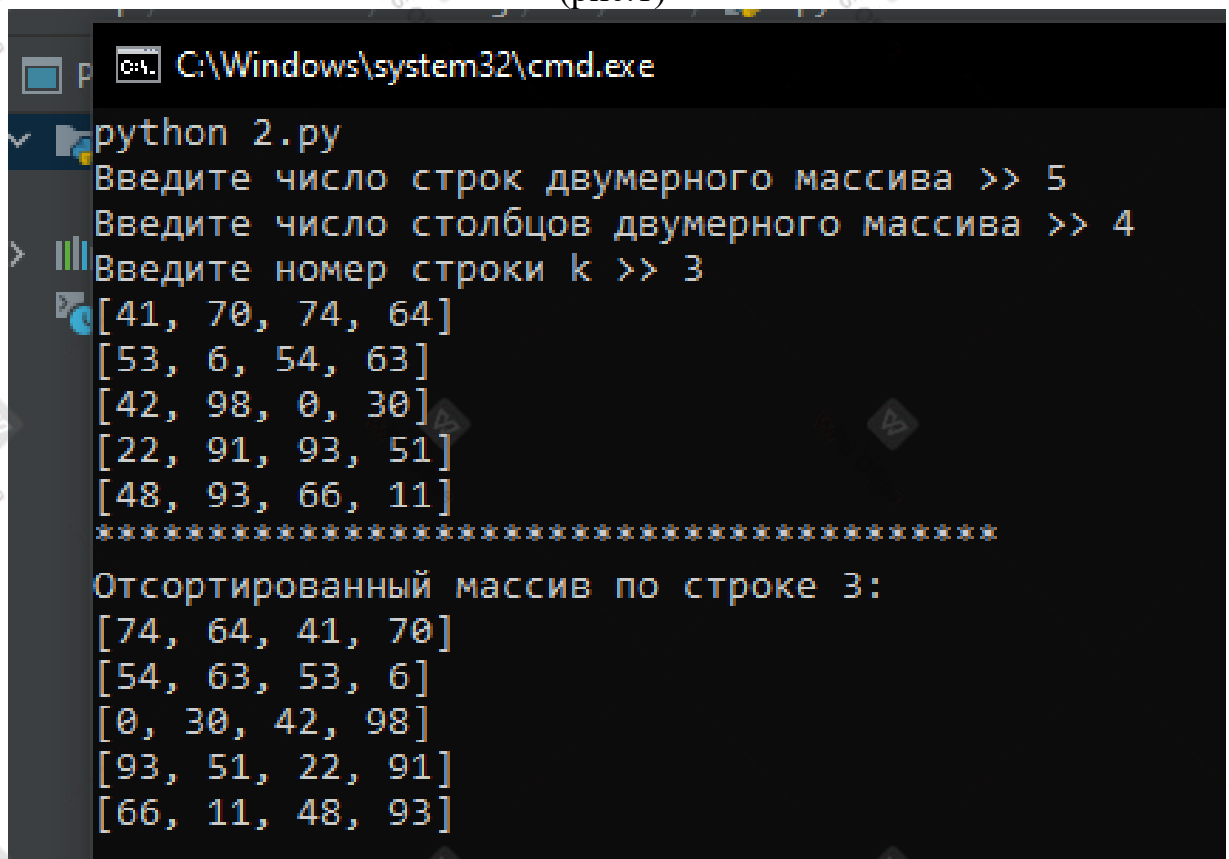
Формулировка:

2. Расположить столбцы матрицы $D[M, N]$ в порядке возрастания элементов k -й строки ($1 \leq k \leq M$).



```
1 from random import *
2
3 # Получаем ширину и высоту матрицы
4 n = int(input('Введите число строк двумерного массива >> '))
5 m = int(input('Введите число столбцов двумерного массива >> '))
6 # Получаем номер строки k
7 k = int(input('Введите номер строки k >> ')) - 1
8
9 # Инициализируем и генерируем матрицу
10 l = []
11 for i in range(n):
12     l.append(list())
13     for j in range(m):
14         l[i].append(randint(0, 100))
15
16 # Вывод матрицы
17 for i in l:
18     print(i)
19
20 print(' ' * 40)
21
22 # Сортировка элементов по возрастанию в строке k и всех остальных элементов
23 for s in range(0, m):
24     for i in range(1, m):
25         if l[k][i] < l[k][i-1]:
26             for j in range(n):
27                 l[j][i], l[j][i-1] = l[j][i-1], l[j][i]
28
29 # Вывод результата
30 print('Отсортированный массив по строке (k + 1):')
31 for i in l:
32     print(i)
```

(рис.1)



```
C:\Windows\system32\cmd.exe
python 2.py
Введите число строк двумерного массива >> 5
Введите число столбцов двумерного массива >> 4
Введите номер строки k >> 3
[41, 70, 74, 64]
[53, 6, 54, 63]
[42, 98, 0, 30]
[22, 91, 93, 51]
[48, 93, 66, 11]
*****
Отсортированный массив по строке 3:
[74, 64, 41, 70]
[54, 63, 53, 6]
[0, 30, 42, 98]
[93, 51, 22, 91]
[66, 11, 48, 93]
```

Пример работы программы (рис.2)

```

1  from random import *
2
3  # Получаем ширину и высоту матрицы
4  n = int(input('Введите число строк двумерного массива >> '))
5  m = int(input('Введите число столбцов двумерного массива >> '))
6  # Получаем номер строки k
7  k = int(input('Введите номер строки k >> ')) - 1
8
9  # Инициализируем и генерируем матрицу
10 l = []
11 for i in range(n):
12     l.append(list())
13     for j in range(m):
14         l[i].append(randint(0, 100))
15
16 # Вывод матрицы
17 for i in l:
18     print(i)
19
20 print('*' * 40)
21
22 # Сортировка элементов по возрастанию в строке k и всех остальных элементов
23 for s in range(0, m):
24     for i in range(1, m):
25         if l[k][i] < l[k][i-1]:
26             for j in range(n):
27                 l[j][i], l[j][i-1] = l[j][i-1], l[j][i]
28
29 # Вывод результата
30 print(f'Отсортированный массив по строке {k + 1}:')
31 for i in l:
32     print(i)
33

```

(рис.3)

Код программы:

```
from random import *

# Получаем ширину и высоту матрицы
n = int(input('Введите число строк двумерного массива >> '))
m = int(input('Введите число столбцов двумерного массива >> '))
# Получаем номер строки k
k = int(input('Введите номер строки k >> ')) - 1

# Инициализируем и генерируем матрицу
l = []
for i in range(n):
    l.append(list())
    for j in range(m):
        l[i].append(randint(0, 100))

# Вывод матрицы
for i in l:
    print(i)

print('*' * 40)

# Сортировка элементов по возрастанию в строке k и всех остальных элементов
for s in range(0, m):
    for i in range(1, m):
        if l[k][i] < l[k][i-1]:
            for j in range(n):
                l[j][i], l[j][i-1] = l[j][i-1], l[j][i]

# Вывод результата
print(f'Отсортированный массив по строке {k + 1}:')
for i in l:
    print(i)
```

Вывод:

Выполнив данную лабораторную работу я узнал способ описания двумерного массива, способы ввода элементов двумерного массива; научился вводить массивы, получать списки через присваивание конкретных значений, применять функции; овладел основными навыками создания программ обработки двумерных массивов.

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