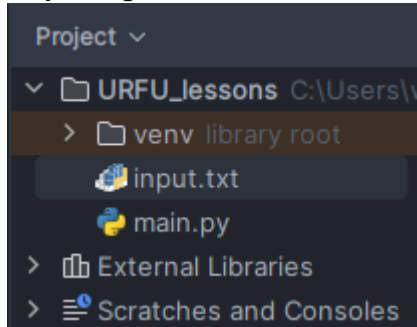


Лабораторная работа 7

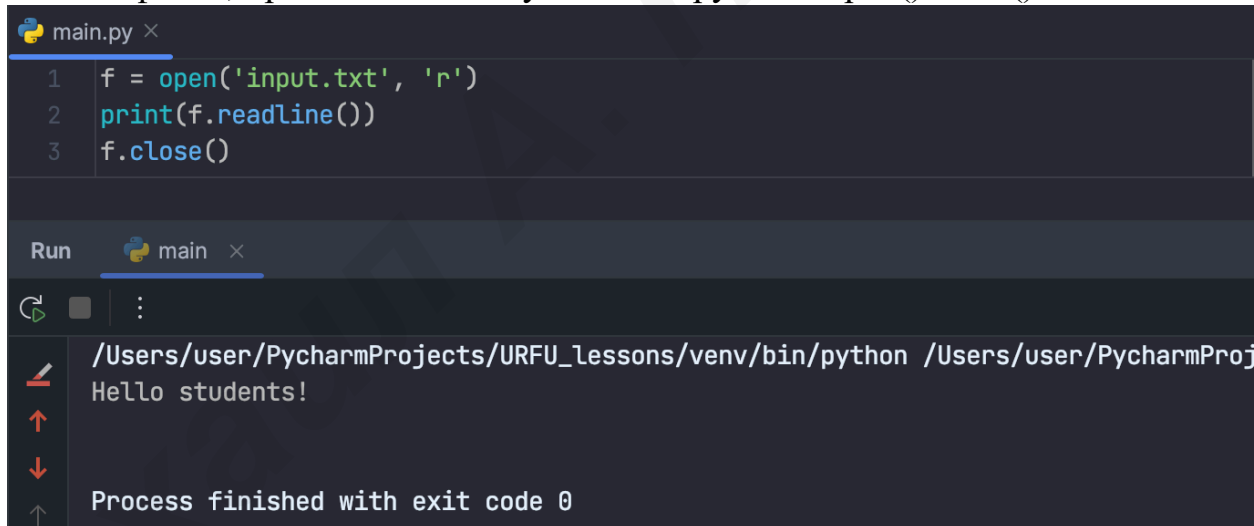
ТЕМА 7. Работа с файлами (ввод, вывод)

Лабораторные задания:

- 1) Составьте текстовый файл и положите его в одну директорию с программой на Python. Текстовый файл должен состоять минимум из двух строк.



- 2) Напишите программу, которая выведет только первую строку из вашего файла, при этом используйте конструкцию `open()/close()`.



- 3) Напишите программу, которая выведет все строки из вашего файла в массиве, при этом используйте конструкцию `open()/close()`.

```
main.py ×
1 f = open('input.txt', 'r')
2 print(f.readlines())
3 f.close()

Run main ×

/Users/user/PycharmProjects/URFU_lessons/venv/bin/python /Users/user/PycharmProj
['Hello students!\n', 'Lets talk about work with files on Python']

Process finished with exit code 0
```

- 4) Напишите программу, которая выведет все строки из вашего файла в массиве, при этом используйте конструкцию with open().

```
main.py ×
1 with open('input.txt') as f:
2     print(f.readlines())
3

Run main ×

/Users/user/PycharmProjects/URFU_lessons/venv/bin/python /Users/user/PycharmProj
['Hello students!\n', 'Lets talk about work with files on Python']

Process finished with exit code 0
```

- 5) Напишите программу, которая выведет каждую строку из вашего файла отдельно, при этом используйте конструкцию with open().

```
main.py × input.txt
1 with open('input.txt') as f:
2     for line in f:
3         print(line)

Run main ×

C:\Users\wh1tly337\PycharmProjects\URFU_lessons\venv\Scripts\python.exe C:\Users\w
Hello students!

Lets talk about work with files on Python

Process finished with exit code 0
```

- 6) Напишите программу, которая будет добавлять новую строку в ваш файл, а потом выведет полученный файл в консоль. Вывод можно осуществлять любым способом. Обязательно проверьте сам файл, чтобы изменения в нем тоже отображались.

The screenshot displays the PyCharm IDE interface. At the top, two tabs are open: 'main.py' and 'input.txt'. The 'main.py' tab is active, showing a Python script with six lines of code. The script uses the 'with open' context manager to append a new line to 'input.txt' and then reads the entire file back to print its contents. Below the editor, the 'Run' toolbar is visible, with the 'Run' button (a green play icon) highlighted. The 'Run' console shows the command executed: `/Users/user/PycharmProjects/URFU_lessons/venv/bin/python /Users/user/PycharmProjects/URFU_L`. The output of the script is displayed in the console: `['Hello students!\n', 'Lets talk about work with files on Python\n', 'Im additional line']`. Below the console, the 'Process finished with exit code 0' message is shown. At the bottom, the 'input.txt' tab is active, showing the contents of the file after execution: `Hello students!`, `Lets talk about work with files on Python`, and `Im additional line`.

```
main.py × input.txt
1 with open('input.txt', 'a+') as f:
2     f.write('\nIm additional line')
3
4 with open('input.txt', 'r') as f:
5     result = f.readlines()
6     print(result)

Run main ×
/Users/user/PycharmProjects/URFU_lessons/venv/bin/python /Users/user/PycharmProjects/URFU_L
['Hello students!\n', 'Lets talk about work with files on Python\n', 'Im additional line']
Process finished with exit code 0

main.py × input.txt ×
1 Hello students!
2 Lets talk about work with files on Python
3 Im additional line
4
5
6
7

Run main ×
/Users/user/PycharmProjects/URFU_lessons/venv/bin/python /Users/user/PycharmProjects/URFU_L
['Hello students!\n', 'Lets talk about work with files on Python\n', 'Im additional line']
Process finished with exit code 0
```

- 7) Напишите программу, которая переписет всю информацию, которая была у вас в файле до этого, например напишет любые данные из произвольно вами составленного списка. Также не забудьте проверить что измененная вами информация сохранилась в файле.

```
main.py x input.txt
1 lines = ['one', 'two', 'three']
2 with open('input.txt', 'w') as f:
3     for line in lines:
4         f.write('\nCycle run ' + line)
5     print('Done!')
6

Run main x
/Users/user/PycharmProjects/URFU_lessons/venv/bin/python /Users/user/PycharmProjects/URFU_lessons/main.py
Done!
Process finished with exit code 0
```

До:

```
main.py input.txt x
1 Hello students!
2 Lets talk about work with files on Python
3 Im additional line
```

После:

```
main.py input.txt x
1
2 Cycle run one
3 Cycle run two
4 Cycle run three
```

- 8) Выберите любую папку на своем компьютере, имеющую вложенные директории. Выведите на печать в терминал ее содержимое, как и всех подкаталогов при помощи функции `print_docs(directory)`.

```
main.py x
1 import os
2
3
4 1 usage
5 def print_docs(directory):
6     all_files = os.walk(directory)
7     for catalog in all_files:
8         print(f'Папка {catalog[0]} содержит:')
9         print(f'Директории: {", ".join([folder for folder in catalog[1]])}')
10        print(f'Файлы: {", ".join([file for file in catalog[2]])}')
11        print('-' * 40)
12
13 print_docs('/Library/User Pictures/Flowers')
```

```
Run main x
/usr/local/bin/python3.11 /Users/user/PycharmProjects/URFU_lessons/main.py
Папка /Library/User Pictures/Flowers содержит:
Директории:
Файлы: Red Rose.tif, Dandelion.tif, Dahlia.tif, Sunflower.tif, Whiterose.tif, Pop
```

9) Документ «input.txt» содержит следующий текст:

Приветствие

Спасибо

Извините

Пожалуйста

До свидания

Ты готов?

Как дела?

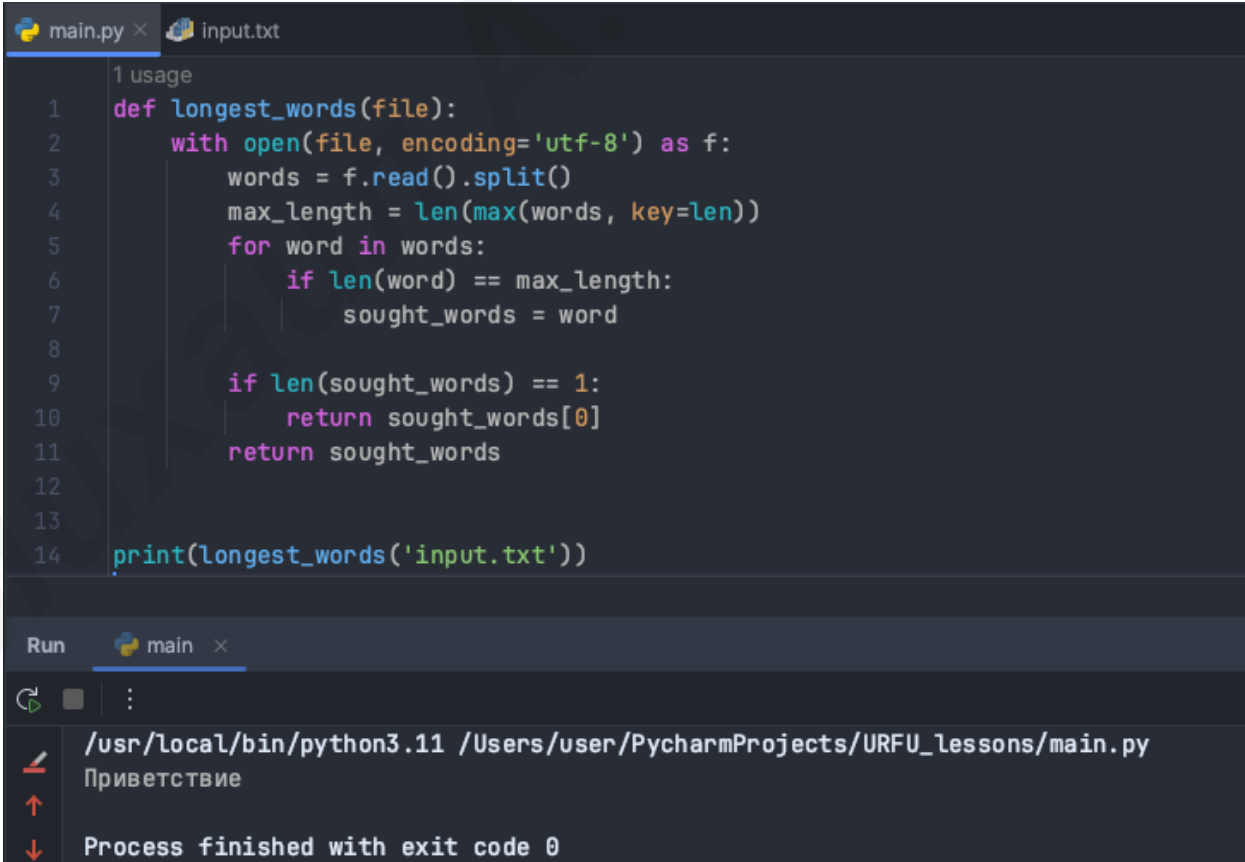
С днем рождения!

Удача!

Я тебя люблю.

Требуется реализовать функцию, которая выводит слово, имеющее максимальную длину (или список слов, если таковых несколько).

Проверьте работоспособность программы на своем наборе данных



The screenshot shows a Python IDE with two tabs: 'main.py' and 'input.txt'. The 'main.py' tab is active, displaying the following code:

```
1 usage
2 def longest_words(file):
3     with open(file, encoding='utf-8') as f:
4         words = f.read().split()
5         max_length = len(max(words, key=len))
6         for word in words:
7             if len(word) == max_length:
8                 sought_words = word
9
10        if len(sought_words) == 1:
11            return sought_words[0]
12        return sought_words
13
14 print(longest_words('input.txt'))
```

Below the code editor, there is a 'Run' button and a terminal window. The terminal shows the output of the program:

```
/usr/local/bin/python3.11 /Users/user/PycharmProjects/URFU_lessons/main.py
Приветствие
Process finished with exit code 0
```

10) Требуется создать csv-файл «rows_300.csv» со следующими столбцами:

- № - номер по порядку (от 1 до 300);

- Секунда – текущая секунда на вашем ПК;
- Микросекунда – текущая миллисекунда на часах.

Для наглядности на каждой итерации цикла искусственно приостанавливайте скрипт на 0,01 секунды.

The screenshot shows a Python IDE with a file named `main.py` open. The code in the file is as follows:

```

1 import csv
2 import datetime
3 import time
4
5 with open('rows_300.csv', 'w', encoding='utf-8', newline='') as f:
6     writer = csv.writer(f)
7     writer.writerow(['№', 'Секунда', 'Микросекунда'])
8     for line in range(1, 301):
9         writer.writerow([line, datetime.datetime.now().second,
10                          datetime.datetime.now().microsecond])
11         time.sleep(0.01)

```

Below the code editor, the 'Run' button is visible, and the output console shows the command executed: `/usr/local/bin/python3.11 /Users/user/PycharmProjects/URFU_lessons/main.py`. The status message indicates: `Process finished with exit code 0`.

The screenshot shows a CSV file named `rows_300.csv` open in the IDE. The file contains 20 rows of data, with columns labeled '№', 'Секунда', and 'Микросекунда'. The data is as follows:

	№	Секунда	Микросекунда
1	1	20	758162
2	2	20	769149
3	3	20	781675
4	4	20	792803
5	5	20	805321
6	6	20	816511
7	7	20	829035
8	8	20	841580
9	9	20	854094
10	10	20	866603
11	11	20	878829
12	12	20	891340
13	13	20	903849
14	14	20	916355
15	15	20	928864
16	16	20	941373
17	17	20	953310
18	18	20	965820
19	19	20	978330
20	20	20	990841

