Лабораторная работа №6

По дисциплине: Основы программной инженерии Гробова Софья

ПИЖ-б-о-20-1

Программа task1.py, код и результат работы

Задача: Дано предложение. Напечатать все его буквы «и».

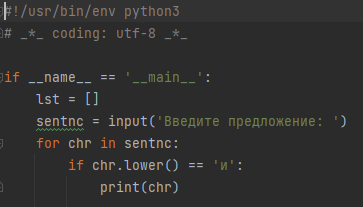


Рисунок 6.1 – Код программы task1.py

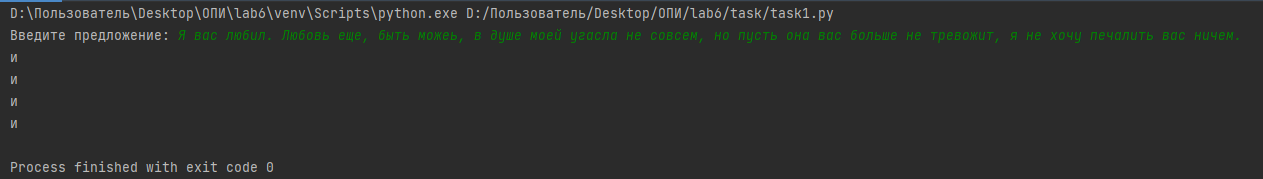


Рисунок 6.2 – Результат работы программы task1.py

Программа task2.py, код и результат работы

Задача: Дано предложение. Определите количество букв н, предшествующих первой запятой предложения. Рассмотреть два случая:

- известно, что запятые в предложении есть;

- запятых в предложении может не быть.

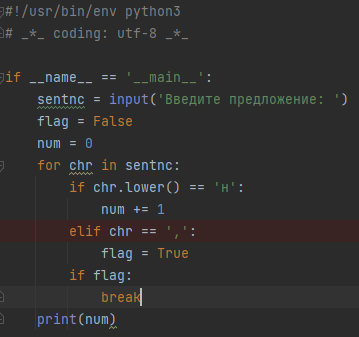


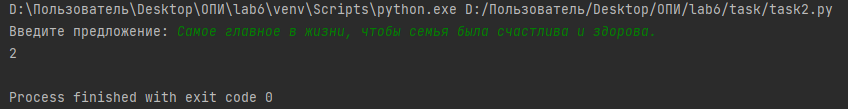
Рисунок 6.3 – Код программы task2.py

Рисунок 6.4 – Результат работы программы task2.py

Программа task3.py, код и результат работы

Задача: Дано предложение. Удалить из него все повторяющиеся буквы, оставив их первые вхождения, т.е. в слове должны остаться только различные буквы.

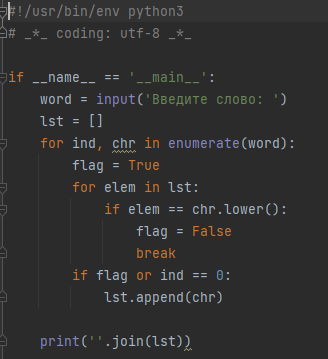


Рисунок 6.5 – Код программы task3.py

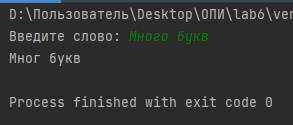


Рисунок 6.6 – Результат работы программы task3.py

Вопросы для защиты работы

1. Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации.
2. Строки в апострофах и кавычках, экранированные последовательности – служебные символы, “Сырые” строки, строки в тройных апострофах или кавычах.
3. Сложение, умножение. Строковых функций много, вот некоторые из них: len() – длина строки

str() – изменяет типо объекта на string

1. <название строковой переменной> [число от 0 до длины строки - 1]
2. Если s это строка, выражение формы s[m:n] возвращает часть s , начинающуюся с позиции m , и до позиции n , но не включая позицию:
3. Так их легче представить в памяти.
4. s.istitle()
5. if s1 in s2
6. s.find(s2)
7. len(s)
8. s.count(<char>)
9. Они позволяют проще формировать строки. Пример: s = f”Ваш id = {id}”
10. s.find(<sub>[, <start>[,<end>]]) 14.” Ваш id = {}”.format(id)
11. s.isdigit()
12. 'foo.bar.baz.qux'.rsplit(sep='.')
13. s.islower()
14. s[0].islower()
15. Нет, можно только преобразовать число в строку и уже его прибавить. 20. s = s[::-1]
16. ‘-‘.join[<iterable>]
17. Верхний – s.upper(), нижний – s.lower()
18. s[0].upper() s[len(s)-1].upper()
19. s.isupper()
20. В случае, если надо сохранить символы конца стркои.
21. s.replace(“что менять”, “на что менять”)
22. string.endswith(<suffix>[, <start>[, <end>]]) – заканчивается, string.startswith(<suffix>[, <start>[,

<end>]]) – начинается.

1. s.isspace()
2. Будет получена копия строки, состоящая из 3 исходных.
3. s.title()
4. string.partition(<sep>) делит строку на основе разделителя. s.partition(<sep>) отделяет от s подстроку длиной от начала до первого вхождения <sep> . Возвращаемое значение представляет собой кортеж из трех частей:

Часть s до <sep> Разделитель <sep> Часть s после <sep>

1. Индекс последнего вхождения подстроки в строку.