

Лабораторная работа №7

По дисциплине: Основы программной инженерии

Духно Михаил

ПИЖ-6-о-20-1

Программа task1.py, код и результат работы

Задача: Дано предложение. Составить программу, которая печатает «столбиком» все вхождения в предложение некоторого символа.

```
text = input('Write text: ')
sign = input('Write sign: ')
text = text.lower()

textArr = text.split()

for word in textArr:
    for ch in word:
        if ch == sign:
            print(word)
```

Рисунок 6.1 – Код программы task1.py

```
C:\Users\я\AppData\Local\Programs\Py
Write text: чебурашка чебурахнулся
Write sign: н
чебурахнулся

Process finished with exit code 0
```

Рисунок 6.2 – Результат работы программы task1.py

Программа task2.py, код и результат работы

Задача: Дано предложение. Определить порядковые номера первой пары одинаковых соседних символов. Если таких символов нет, то должно быть напечатано соответствующее сообщение.

```
text = input('Write text: ')
text = text.lower()

for ind, ch in enumerate(text):
    if ind != 0:
        if ch == text[ind - 1]:
            print(f'Sign "{ch}" in positions {ind} and {ind + 1}')
```

Рисунок 6.3 – Код программы task2.py

```

C:\Users\я\AppData\Local\Programs\Python\Python39\python.
Write text: программа разработана для разнообразия
Sign "o" in positions 30 and 31

Process finished with exit code 0

```

Рисунок 6.4 – Результат работы программы task2.py

Программа task3.py, код и результат работы

Задача: Дано предложение. Удалить из него все буквы о, стоящие на нечетных местах.

```

text = input('Write text: ')
text = text.lower()
resText = ''

for ind, ch in enumerate(text):
    if (ch == 'o' and ind % 2 == 0) or (ch == 'o' and ind % 2 == 0):
        continue
    else:
        resText = resText + ch

print(f'Reformed text: {resText}')

```

Рисунок 6.5 – Код программы task3.py

```

C:\Users\я\AppData\Local\Programs\Python\Python3
Write text: oaoaoaoaoaoaoaoaoaoaoaoaoaoaoaoaoaoao
Reformed text: aaaaaoaoaoaoaoaoaoaoaoaoaoaoaoaoa
Process finished with exit code 0

```

Рисунок 6.6 – Результат работы программы task3.py

Программа hardLevel.py, код и результат работы

Задача: Даны два слова. Напечатать только те буквы слов, которые встречаются в обоих словах лишь один раз. Например, если заданные слова процессор и информация, то ответом должно быть: п е ф м а я.

```

words = input('Write two words: ')
words = words.replace(' ', '')
words = words.lower()
resChars = ''

for ind, ch in enumerate(words):
    if findChar(words, ch, ind):
        continue
    else:
        resChars = resChars + ch + ' '

print(f'Result: {resChars}')

```

Рисунок 6.7 – Код программы hardLevel.py

```
C:\Users\я\AppData\Local\Programs\Python\Python39\py
Write two words: ЧЕБУРАШКА ПИШЕТ ПОКОПАДНЫЕ ТОРТЫКИ
Result: ч у ю д н ы
Process finished with exit code 0
```

Рисунок 6.8 – Результат работы программы hardLevel.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 кавычках.
3. Сложение, умножение. Строковых функций много, вот некоторые из них:
len() – длина строки
str() – изменяет тип объекта на string
4. <название строковой переменной> [число от 0 до длины строки - 1]
5. Если s это строка, выражение формы s[m:n] возвращает часть s , начинающуюся с позиции m , и до позиции n , но не включая позицию:
6. Так их легче представить в памяти.
7. s.istitle()
8. if s1 in s2
9. s.find(s2)
10. len(s)
11. s.count(<char>)
12. Они позволяют проще формировать строки. Пример: s = f”Ваш id = {id}”
13. s.find(<sub>[, <start>[, <end>]])
- 14.” Ваш id = {}”.format(id)
15. s.isdigit()
16. 'foo.bar.baz.qux'.rsplit(sep='.')
17. s.islower()
18. s[0].islower()
19. Нет, можно только преобразовать число в строку и уже его прибавить.
20. s = s[::-1]
21. '-'.join[<iterable>]
22. Верхний – s.upper(), нижний – s.lower()
23. s[0].upper() s[len(s)-1].upper()
24. s.isupper()
25. В случае, если надо сохранить символы конца строки.
26. s.replace(“что менять”, “на что менять”)
27. string.endswith(<suffix>[, <start>[, <end>]]) – заканчивается, string.startswith(<suffix>[, <start>[, <end>]]) – начинается.
28. s.isspace()
29. Будет получена копия строки, состоящая из 3 исходных.
30. s.title()
31. 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>
32. Индекс последнего вхождения подстроки в строку.