

КУРС «ИНФОРМАТИКА»

Часть 7. Подготовка к контрольной работе № 3
(символы и строки)

2021 – 2022 УЧЕБНЫЙ ГОД

Алфавит задач на строки

```
// если в задании не сказано иное, то слова состоят
// из строчных латинских букв
lat = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

```
// если в задании слова состоят ТОЛЬКО из латинских и/или
// русских букв, то необходимо задавать алфавит
lat = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
kir='абвгдеёжзийклмнопрстуфхцчщъыьэюя
АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧЩЪЫЬЭЮЯ'
alph= lat + kir
```

brain – слово только из латинских букв

мозг – слово только из русских букв

слабыйbrain – неподходящее слово

brainспит – неподходящее слово

Алфавит задач на строки

Пример входной строки:

умныйbrain слыrs at night, и не пишет stupid%^4программу.

Количество слов, состоящих только из латинских букв – 3
(at, night, stupid)

Количество слов, состоящих только из русских букв – 4
(и, не, пишет, программу)

Алгоритм проверки:

1. Задать алфавит (латиница + кириллица)
2. Получить слово, состоящее из всех букв алфавита
(умныйbrain, слыrs – это тоже слова)
3. Проверить каждую букву на принадлежность нужному
множеству символов

Способы задания алфавита

- 1) # обычное перечисление букв
lat = 'abcdefghijklmnopqrstuvwxyz '
kir='абвгдеёжзийклмнопрстуфхцщъыьэюя '
- 2) # использование констант из модуля string
import string
#Константное перечисление букв из набора ASCII в верхнем регистре
string.ascii_uppercase
#Константное перечисление букв из набора ASCII в нижнем регистре
string.ascii_lowercase
#Константное перечисление букв из набора ASCII в верхнем и нижнем регистре
string.ascii_letters
- 3) lat=''
for i in range (ord("A"),ord("Z")+1):
lat = lat + chr(i)

Пример задачи № 1.1

```
// Дана строка символов. Заменить в строке каждое слово,  
// начинающееся и заканчивающееся одной и той же буквой на  
// слово-"перевертыш" (исходное слово в обратном порядке),  
// посчитать количество таких замен.
```

```
// например, для строки  
alfa demo dead
```

```
// программа должна вывести
```

Строка после замены:

afla demo daed

Количество замен: 2

Пример задачи № 1.1

```
# начало программы
lat =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

# получение первой и последней позиции слова в строке
def getWord (s, fp):
    while (fp < len(s)) and (not (s[fp] in lat)):
        fp = fp+1
    lp = fp
    while (lp < len(s)) and (s[lp] in lat):
        lp = lp+1
    lp = lp-1
    return [fp, lp]
```

Пример задачи № 1.1

```
def alg(s):  
    i = 0  
    k = 0  
    while i < len(s):  
        [fp, lp] = getWord(s, i)  
        word = s[fp:lp+1]  
        if s[fp] == s[lp]:  
            revWord = word[::-1]  
            k += 1  
            s = s[:fp] + revWord + s[lp+1:]  
        i = lp + 1  
    return [s, k]
```

Пример задачи № 1.1

```
s = input('Введите строку: ')\n[s, k] = alg(s)\nprint("Строка после замены: ", s)\nprint("Количество замен: ", k)
```


Пример задачи № 1.2

```
// Дана строка символов. Заменить в строке каждое слово,  
// начинающееся заданным фрагментом, на слово, заданное  
// пользователем, посчитать количество таких замен.
```

```
// например, для строки  
alfa demo alldead ksal alal
```

```
// фрагмента al  
// и слова ddd
```

```
// программа должна вывести
```

Строка после замены:

ddd demo ddd ksal ddd

Количество замен: 3

Пример задачи № 1.2

```
# начало программы
lat =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

# Получение первой и последней позиции слова в строке
def getWord (s, fp):
    while (fp < len(s)) and (not (s[fp] in lat)):
        fp = fp+1
    lp = fp
    while (lp < len(s)) and (s[lp] in lat):
        lp = lp+1
    lp = lp-1
    return [fp, lp]
```

Пример задачи № 1.2

```
def alg(s, fr, z):  
    i=0; k=0;  
    while i<len(s):  
        [fp, lp] = getWord(s, i)  
        word = s[fp:lp + 1]  
        if word.find(fr)==0:  
            s= s[:fp] + z + s[lp+1:]  
            i=fp+len(z)  
            k=k+1  
        else:  
            i=lp+1  
    return [s,k]
```

alfa demo alldead ksal alal

ddd demo ddd ksal ddd

Пример задачи № 1.2

```
s = input('Введите строку: ')
fr = input('Введите фрагмент: ')
z = input('Введите слово: ')
[s, k] = alg(s, fr, z)
print("Строка после замены: ", s)
print("Количество замен: ", k)
```

Пример задачи № 2.1

```
// Дана строка символов. Продублировать все слова,  
// начинающиеся с гласной латинской буквы,  
// посчитать количество слов после дублирования.
```

```
// например, для строки  
alfa demo elad ksal lal
```

```
// программа должна вывести
```

Строка после дублирования:

alfa alfa demo elad elad ksal lal

Количество слов: 7

Пример задачи № 2.1

```
# начало программы
lat =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
gl = 'AEIOUY'
# Получение первой и последней позиции слова в строке
def getWord (s, fp):
    while (fp < len(s)) and (not (s[fp] in lat)):
        fp = fp+1
    lp = fp
    while (lp < len(s)) and (s[lp] in lat):
        lp = lp+1
    lp = lp-1
    return [fp, lp]
```

Пример задачи № 2.1

```
def alg(s):  
    i=0; k=0;  
    while i<len(s):  
        [fp, lp] = getWord(s, i)  
        word = s[fp:lp + 1]  
        if word[0].upper() in gl:  
            s= s[:fp] + word + ' ' + word + s[lp+1:]  
            i = lp + len(word) + 2  
            k = k + 1  
        else:  
            i = lp+1  
            k = k + 1  
    return [s,k]
```

Пример задачи № 2.1

```
s = input('Введите строку: ')\n[s, k] = alg(s)\nprint("Строка после дублирования: ", s)\nprint("Количество слов после дублирования: ", k)
```


Пример задачи № 2.2

```
// Дана строка символов. Удалить все слова,  
// заканчивающиеся заданным фрагментом,  
// посчитать количество удаленных слов.
```

```
// например, для строки  
alal demo elad ksal lal
```

```
// и фрагмента al
```

```
// программа должна вывести
```

Строка после удаления:

demo elad

Количество удаленных слов: 3

Пример задачи № 2.2

```
# начало программы
lat =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

# Получение первой и последней позиции слова в строке
def getWord (s, fp):
    while (fp < len(s)) and (not (s[fp] in lat)):
        fp = fp+1
    lp = fp
    while (lp < len(s)) and (s[lp] in lat):
        lp = lp+1
    lp = lp-1
    return [fp, lp]
```

Пример задачи № 2.2

```
def alg(s, fr):  
    i=0; k=0;  
    while i<len(s):  
        [fp, lp] = getWord(s, i)  
        word = s[fp:lp + 1]  
        if word.endswith(fr):  
            s= s[:fp] + s[lp+1:]  
            k = k + 1  
        else:  
            i = lp+1  
  
    return [s, k]
```

Пример задачи № 2.2

```
s = input('Введите строку: ')
fr = input('Введите фрагмент: ')
[s, k] = alg(s, fr)
print("Строка после удаления: ", s)
print("Количество удаленных слов: ", k)
```

Дополнительные задачи

для подготовки к КР № 3

1 Дана строка символов. Заменить в строке каждое слово, начинающееся с заданного фрагмента, на то же слово, записанное в обратном порядке, посчитать количество таких замен.

2 Дана строка символов. Продублировать каждое слово, содержащее заданную букву, посчитать количество всех слов после выполнения дублирования.

1 Дана строка символов. Заменить в строке каждое слово, состоящее только из английских букв, на слово, заданное пользователем, посчитать количество таких замен.

2 Дана строка символов. Удалить каждое второе слово, равное заданному слову, посчитать количество удаленных слов

КУРС «ИНФОРМАТИКА»

Часть 8. Памятка по Python № 3

2021 – 2022 УЧЕБНЫЙ ГОД

Список функций

Название функции	Описание	Пример использования
<pre>string.replace(old , new) string.replace(old , new, max)</pre>	возвращает копию строки, в которой подстрока old будет заменена новой подстрокой new. Параметр max – дополнительный аргумент, заменяется только первые max вхождений	<pre>str = "Первое, второе, третье" newStr= str.replace("второе", "2") print(newStr) # Первое, 2, третье</pre>
<pre>string.find(fr) string.find(fr, start) string.find(fr, start, end)</pre>	Функция находит первое вхождение переданного фрагмента(fr) и возвращает его индекс, если найден и -1 в противном случае. start – индекс начала среза в исходной строке, в котором требуется отыскать подстроку. end – индекс конца среза в исходной строке, в котором требуется отыскать подстроку	<pre>str = "Первое, второе, третье" index= str.find("в", 6, 15) print(index) #8</pre>

Список функций

Название функции	Описание	Пример использования
<pre>str.startswith(fr) str.startswith(fr, start) str.startswith(fr, start, end)</pre>	<p>Возвращает true, если строка начинается указанным фрагментом(fr).</p> <p>start – позиция (индекс символа), с которой следует начать поиск.</p> <p>end – позиция (индекс символа), на которой следует завершить поиск.</p>	<pre>str = "минотавр" index= str.startswith("вр") print(index) #false</pre>
<pre>str.endswith(fr) str.endswith(fr, start) str.endswith(fr, start, end)</pre>	<p>Возвращает true, если строка оканчивается указанным фрагментом(fr).</p> <p>start – позиция (индекс символа), с которой следует начать поиск.</p> <p>end – позиция (индекс символа), на которой следует завершить поиск.</p>	<pre>str = "минотавр" index= str.endswith("вр") print(index) #true</pre>

Список функций

Название функции	Описание	Пример использования
<code>string.upper()</code>	Возвращает копию исходной строки с символами приведёнными к верхнему регистру	<pre>str = "большие буквы" strNew = str.upper() print(strNew) #БОЛЬШИЕ БУКВЫ</pre>
<code>string.lower()</code>	Возвращает копию исходной строки с символами приведёнными к нижнему регистру	<pre>str = "МАЛЕНЬКИЕ БУКВЫ" strNew = str.lower() print(strNew) #маленькие буквы</pre>