

Задание 1.

Создайте функцию, работающую по следующему алгоритму:

Выводятся все буквы алфавита: абвгдеёжзийклмнопрстуфхцчшщъыьэюя

Пользователь вводит произвольную строку, символы этой строки сравниваются с символами строки алфавита. Вывести символы, использовавшиеся в строке пользователя вперёд и убрать их из алфавита, а затем, вывести и оставшиеся буквы из алфавита.

Входные данные:

```
alpha('пайтон')
```

Выходные данные:

```
а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я
п а й т о н б в г д е ё ж з и к л м р с у ф х ц ч ш щ ъ ы ь э ю я
```

Задание 2.

Создайте функцию, принимающую на вход название месяца, год и количество дней. Функция должна создать календарь, по переданным ей значениям и вывести его в привычном виде. Все три параметра могут быть произвольные и не зависеть от действительности. Но, количество дней в неделе должно оставаться 7.

Входные данные:

```
create_calendar('Randomner', 2045, 23)
```

Выходные данные:

```
calendar: Randomner 2045
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23
```

Задание 3.

Написать функцию, принимающую на вход два целых числа. Функция будет выводить последовательность чисел от первого до второго включительно и сумму всех этих чисел в двоичной системе счисления.

Входные данные:

```
bin_sys(3, 6)
```

Выходные данные:

```
11
```

```
100
```

```
101
```

```
110
```

```
сумма: 10010
```

Задание 4.

Есть двумерный массив:

```
field = [['[ ]', '[ ]', '[ ]'],  
         ['[ ]', '[ ]', '[ ]'],  
         ['[ ]', '[ ]', '[ ]']]
```

Реализуйте функцию, которая по переданным координатам заменит элемент [] на *

Функция принимает 3 аргумента: массив, строка и столбец.

Входные данные:

```
begin(field, 1, 2)
```

Выходные данные:

```
[ ] [ ] [ ]
```

```
[ ] [ ] *
```

```
[ ] [ ] [ ]
```

Задание 5.

Создайте функцию, которая принимает 2 аргумента и создаёт некоторое поле: [n1] [n2]

[n3] [n4]

Где n1 – это первый аргумент функции и первое число данного поля. А второй аргумент – это некий шаг, между числами. Он не является обязательным, и если его не применить, то числа выведутся в порядке возрастания.

Последующие числа будут иметь вид: $n2 = n1 + 1 * \text{второй аргумент}$. Где к последующему числу будет прибавляться на 1 единицу больше.

В данной функции, должна быть только одна строка кода.

Входные данные 1:

```
_numbers(1)
```

Выходные данные 1:

```
[1] [2]
```

```
[3] [4]
```

Входные данные 2:

```
_numbers(1, 2)
```

Выходные данные 2:

```
[1] [4]
```

```
[6] [8]
```

Задание 6.

Реализовать функцию с двумя аргументами: текст и искомая буква. Функция будет проверять, сколько раз в строке встречается буква. При этом, регистр не имеет значение.

Входные данные:

```
exam('My name is Sara.', 's')
```

Выходные данные:

```
2
```