

Лабораторная работа №1

- Даны две строки str1 и str2. Программа должна создать новую строку str3 состоящую из первого, среднего и последнего символов строк str1 и str2. Полученный результат выведите в терминал. Например: str1 = «Мама», str2 = «Утром» → «МУмрам».

```
def str_task_9():
    print('str 9 task\n')
    str1 = input("input str1: ")
    str2 = input("input str2: ")

    middle_str1 = len(str1) // 2
    middle_str2 = len(str2) // 2

    new_str = "".join(
        [str1[0],
         str2[0],
         str1[middle_str1],
         str2[middle_str2],
         str1[-1],
         str2[-1]]
    )
    print(new_str)
```

Вывод

```
str 9 task

input str1: mama
input str2: ytrom
mymram

str task_10
```

- Напишите программу переворачивающую строку «Как дела?» без использования циклов. Полученный результат выведите в терминал. Например: «Йо-хо-хо!» → «!ох-ох-ой»

```
def str_task_10():
    print('\nstr task_10\n')
    string = "Как дела?"
    print(string[::-1])
```

Вывод

```
str task_10
```

```
?алед как
```

- Дана строка «В данной главе мы ознакомились с тем, как выполнить установку интерпретатора». Используя пробелы, разбейте ее на части, сформировав список. Полученный результат выведите в терминал.

```
def str_task_11():  
    print('\nstr_task_11\n')  
    string = "В данной главе мы ознакомились с тем, как выполнить  
установку интерпретатора"  
    print(string.split())
```

Вывод

```
str_task_11
```

```
['В', 'данной', 'главе', 'мы', 'ознакомились', 'с', 'тем,', 'как',  
'выполнить', 'установку', 'интерпретатора']
```

- Дан список my_list = [1, 2, 3]. Добавьте в его начало значение 12 и выведите в терминал полученный результат.

```
def list_task_14():  
    print('\nlist_task_14\n')  
    my_list = [1, 2, 3]  
    print("old_my_list", my_list)  
    my_list.insert(0, 12)  
    print("my_list", my_list)
```

Вывод

```
list_task_14
```

```
old_my_list [1, 2, 3]
```

```
my_list [12, 1, 2, 3]
```

- Дан список my_list = [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2]. Посчитайте количество вхождения в него элементов со значением 2 и выведите в терминал

полученный результат. Для решения задачи запрещено использовать циклы.

```
def list_task_17():  
    print('\nlist_task_17\n')  
    my_list = [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2]  
    print('my_list is', my_list)  
    print('count of 2 in them', my_list.count(2))
```

Вывод

```
list_task_17  
  
my_list is [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2]  
count of 2 in them 4
```

- Дан список my_list = [1, -2, 43, -42, 59, 6, 23]. Проверьте входит ли в него элемент со значением 6 и выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы

```
def list_task_20():  
    print('\nlist_task_20\n')  
    my_list = [1, -2, 43, -42, 59, 6, 23]  
    print('my_list is', my_list)  
    print('6 in my_list:', 6 in my_list)
```

Вывод

```
list_task_20  
  
my_list is [1, -2, 43, -42, 59, 6, 23]  
6 in my_list: True
```

Дан список my_list = [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2, 'a', 'b', 2, 5]. На его основе сформируйте новый, состоящий из первого, среднего и последнего элемента первой переменной. Полученный результат выведите в терминал.

```
def list_task_27():  
    print('\nlist_task_27\n')  
    my_list = [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2, 'a', 'b', 2, 5]  
    print('my_list is', my_list)  
    middle_id = len(my_list) // 2  
    new_list = [my_list[0], my_list[middle_id], my_list[-1]]  
    print('new_list is', new_list)
```

Вывод

```
list_task_27
```

```
my_list is [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2, 'a', 'b', 2, 5]  
new_list is [1, 3, 5]
```

- Дан кортеж my_tuple = (1, 2, 3, 1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2). Посчитайте количество вхождения в него элементов со значением 3 и выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы.

```
def tuple_task_6():  
    print('\ntuple_task_6\n')  
    my_tuple = (1, 2, 3, 1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2)  
    print('my_tuple is', my_tuple)  
    print('count of 3 in my_tuple:', my_tuple.count(3))
```

Вывод

```
tuple_task_6
```

```
my_tuple is (1, 2, 3, 1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2)  
count of 3 in my_tuple: 3
```

- Дан кортеж my_tuple = (0, -2, 81, 42, 6, -6, 23). Найдите элемент с минимальным значением и выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы.

```
def tuple_task_8():  
    print('\ntuple_task_8\n')  
    my_tuple = (0, -2, 81, 42, 6, -6, 23)  
    print('my_tuple is', my_tuple)  
    print('min(my_tuple) is', min(my_tuple))
```

Вывод

```
tuple_task_8
```

```
my_tuple is (0, -2, 81, 42, 6, -6, 23)  
min(my_tuple) is -6
```

- Дан кортеж `my_tuple = (0, -2, 81, 42, 6, -6, 23)`. Проверьте входит ли в него элемент со значением `-6` и выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы

```
def tuple_task_9():  
    print('\ntuple_task_9\n')  
    my_tuple = (0, -2, 81, 42, 6, -6, 23)  
    print('my_tuple is', my_tuple)  
    print('-6 in my_tuple:', -6 in my_tuple)
```

Вывод

```
tuple_task_9  
  
my_tuple is (0, -2, 81, 42, 6, -6, 23)  
-6 in my_tuple: True
```

- Дан кортеж `my tuple = (1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2, 'a', 'b', 2, 5)`. Используя срезы сформируйте новый кортеж, в состав которого входят последние 5 элементов `my tuple`. Выведите в терминал полученный результат.

```
def tuple_task_16():  
    print('\ntuple_task_16\n')  
    my_tuple = (1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2, 'a', 'b', 2, 5)  
    print('my_tuple is', my_tuple)  
    print('erased_tuple:', my_tuple[-5:])
```

Вывод

```
tuple_task_16  
  
my_tuple is (1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2, 'a', 'b', 2, 5)  
erased_tuple: (2, 'a', 'b', 2, 5)
```

- Запуск всех программ с целью проверки

```
def main():  
    str_task_9()  
    str_task_10()  
    str_task_11()  
    print('---')  
  
    list_task_14()  
    list_task_17()
```

```
list_task_20()
list_task_27()
print('---')

tuple_task_6()
tuple_task_8()
tuple_task_9()
tuple_task_16()

if __name__ == "__main__":
    main()
```