

Лабораторная работа №2

Мой вариант - 10

Задания на множества:

- 1. Дан кортеж `my_tuple = (1, 2, 5, 6, 7, 6, 2, 5)`. На его основе сформируйте множество `my_set` и выведите в терминал полученный результат.
- 11. Дан список `my_list = [1, 0, 1, 10, 5, 6, 7, 4, 4, 1, 6, 2, 5]`. Сформируйте новый список, содержащий не повторяющиеся элементы, и выведите его в терминал
- 14. Дано множество $A = \{0, 1, 2, 6, 7, 8, 9\}$ и $B = \{1, 3, 6, 10, 9, 21, 5\}$. Найдите разницу $A - B$ и выведите полученный результат в терминал.
- 15. Дано множество $A = \{0, 1, 2, 6, 7, 8, 9\}$ и $B = \{1, 3, 6, 10, 9, 21, 5\}$. Найдите разницу $B - A$ и выведите полученный результат в терминал.

```
def set_task_1():
    print('set_task_1')
    my_tuple = (1, 2, 5, 6, 7, 6, 2, 5)
    print(set(my_tuple))

def set_task_11():
    print('\nset_task_11')
    my_list = [1, 0, 1, 10, 5, 6, 7, 4, 4, 1, 6, 2, 5]

    my_set = set(my_list)

    list_without_equal_elems = list(my_set)

    print(list_without_equal_elems)

def set_task_14_and_15():
    print('\nset_task_14 and 15')
    A = {0, 1, 2, 6, 7, 8, 9}
    B = {1, 3, 6, 10, 9, 21, 5}

    print('A-B=', A-B)
    print('B-A=', B-A)

def main():
    set_task_1()
    set_task_11()
    set_task_14_and_15()
```

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

Вывод

```
set_task_1  
{1, 2, 5, 6, 7}  
  
set_task_11  
[0, 1, 2, 4, 5, 6, 7, 10]  
  
set_task_14 and 15  
A-B= {0, 8, 2, 7}  
B-A= {10, 21, 3, 5}
```

Задания на словари:

- 5. Дан словарь `my_dict = {1: 10, 'a': 5, 'b': -2}`. Получите сумму значений всех его элементов и выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы.
- 7. Дан словарь `my_dict = {1: 10, 'a': 5, 'b': -2}`. Измените значение, хранящееся по ключу 'a' на 10 и выведите в терминал полученный результат.
- 11. Дан список `my_list1 = [1, 2, 3, 4, 9, 7, 4]` и `my_list2 = [2, 13, 4, 8, 7, 6]`. Напишите программу, формирующую словарь, где в качестве ключей выступают элементы первого списка, а в качестве значений - второго. Выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы.
- 16. Дан словарь `my_dict = {'name': 'Alex', 'age': 25, 'salary': 8000}`. Сформируйте список из ключей словаря и выведите в терминал полученный результат. Для решения задачи запрещено использовать циклы.

```
def dict_task_5():  
    print('dict_task_5')  
    my_dict = {1: 10, 'a': 5, 'b': -2}  
    print('sum', sum(my_dict.values()))  
  
def dict_task_7():  
    print('\ndict_task_7')  
    my_dict = {1: 10, 'a': 5, 'b': -2}
```

```

print('old_my_dict', my_dict)
my_dict['a'] = 10
print('new_my_dict', my_dict)

def dict_task_11():
    print('\ndict_task_11')
    my_list1 = [1, 2, 3, 4, 9, 7, 4]
    my_list2 = [2, 13, 4, 8, 7, 6]
    my_dict = dict(zip(my_list1, my_list2))
    print(my_dict)

def dict_task_16():
    print('\ndict_task_16')
    my_dict = {'name': 'Alex', 'age': 25, 'salary': 8000}
    print(list(my_dict.keys()))

def main():
    dict_task_5()
    dict_task_7()
    dict_task_11()
    dict_task_16()

if __name__ == '__main__':
    main()

```

Вывод

Задания на побитовые операции:

- 3. Используя побитовую операцию возведите значение 2 в 5-ю степень и выведите полученный результат в двоичном формате в терминал.
- 6. Дана переменная $z = 0b1011010$. Проверьте установлен ли ее левый бит в единицу или нет и выведите полученный результат в терминал
- 7. Дана переменная $z = 0b1111011$. Установите ее правый бит в ноль и выведите полученный результат в двоичном формате в терминал.
- 15. Дана переменная $z = 174$. Посчитайте количество занимаемых ей бит и выведите полученный результат в терминал

```

def pobit_task_3():
    print("pobit_task_3")
    temp = 2
    temp <<= 5;

```

```

    print(temp)

def pobit_task_6(z: int = 0):
    print("\npobit_task_6")
    print(z, bin(z))
    answer = "yes" if z & 1 else "no"
    print(answer)

def pobit_task_7(z: int = 0b1111011):
    print('\npobit_task_7')
    print(z, bin(z))
    z >>= 1
    z <<= 1
    print(z, bin(z))

def pobit_task_15(z: int = 174):
    print('\npobit_task_15')
    print(z, bin(z))
    count_of_bits = 0
    while z > 0:
        count_of_bits += 1
        z >>= 1
    print('count of bits', count_of_bits)

def main():
    pobit_task_3()
    temp = 0b1011010
    pobit_task_6(temp)
    temp = 0b1011011
    pobit_task_6(temp)

    pobit_task_7()

    pobit_task_15()

if __name__ == '__main__':
    main()

```

Вывод

```

pobit_task_3
64

pobit_task_6
90 0b1011010
no

pobit_task_6

```

```
91 0b1011011
```

```
yes
```

```
pobit_task_7
```

```
123 0b1111011
```

```
122 0b1111010
```

```
pobit_task_15
```

```
174 0b10101110
```

```
count of bits 8
```