

Zadanie 1

```
if __name__ == '__main__':  
    print(odwr('To jest tekst do odwrocenia'))  
    print(most_common_element([2,2,10]))  
    print(most_common_element([]))  
    print(newt(49, 0.1))  
    print(make_alpha_dict('ona i on'))  
    print(flatten([1, [2, 3], [[4, 5], 6]]))  
    print(flatten([[[[2,[[[]],[[]]]]])])
```

aine corwd o d tsket tsej oT

2

None

7.001406475243939

{'o': ['ona', 'on'], 'n': ['ona', 'on'], 'a': ['ona'], 'i': ['i']}

[1, 2, 3, 4, 5, 6]

[2]

Zadanie 2

```
if __name__ == '__main__':  
    print(forall(lambda x: x < 10, [1, 10,11,12]))  
    print(exists(lambda x: x < 10, [1, 10,11,12]))  
    print(atleastn(2, lambda x: x < 10, [1, 10,11,12]))  
    print(atmost(2, lambda x: x < 10, [1, 10,11,12]))  
    print(forall(lambda x: x < 10, range(0,100)))
```

False

True

False

True

False

Zadanie 3

```
if __name__ == '__main__':
    pg1 = PasswordGenerator(10, [], 3)
    pg2 = PasswordGenerator(5, ['a', 'b', 'c', '5', '1', '2'], 0)
    pg3 = PasswordGenerator(25, ['a', 'b', 'c', '5', '1', '2'], 2)
    pg4 = PasswordGenerator(10, ['$', '!', '#', '5'], 3)

    def test(tests):
        for num, gen in enumerate(tests):
            print(f'{num+1}:')
            for i in gen: print(f'\t{i}')
            print()
        return

    test([pg1, pg2, pg3, pg4])
```

```
1:
    FjFyFsVNud
    IsEqBBxsnv
    GUVjSFayWt

2:

3:
    55aba2ac5ac1aa51a125c12aa
    ac52b221bc515a1cba221ca25

4:
    55555#!555
    $$$5$5!$!!
    55#5$!$5##
```

Zadanie 4, 5

```
if __name__ == '__main__':
    fgen = make_generator(fib)
    print([i for i in fgen(10)])

    gen1 = make_generator(lambda n: 2*n)
    gen2 = make_generator(lambda n: 3**n * 2)
    gen3 = make_generator_mem(fib)

    print([i for i in gen1(5)])
    print([i for i in gen2(5)])
    print([i for i in gen3(10)])
```

```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
[2, 4, 6, 8, 10]
[6, 18, 54, 162, 486]
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

Zadanie 6

```
@log(DEBUG)
def foo(n):
    return n * 2

@log(DEBUG)
class Test:
    def __init__(self):
        return

if __name__ == '__main__':
    Test()
    foo(2)
    pass
```

```
Zainicjowano instancje klasy Test  
Czas wywołania: 1746289164.1155806  
Czas trwania: 1.6689300537109375e-06  
Wartość zwracana: foo((2,), {}) -> 4
```