1

def upper(t:str):

x = ""

for I in t:

if i.isupper():

x += i + " "

if x:

print(x[:-1])

else:

print("Нет заглавных букв")

upper('PriVet')

2

def punct(txt):

count = 0

for char in txt:

if char in '!?.,()':

count += 1

print(count)

punct('(Как дела?)')

3

def create\_figure(x, y):

for \_ in range(y):

print('\*' \* x)

create\_figure (5, 3)

4

def doub(s):

result = ''

for i in s:

result += i \* 2

print(result)

double('строка')

5

def construct(scraps, humans, cars, trees):

sets\_count = min(scraps // 72, humans // 4, cars // 2, trees // 7)

return sets\_count

result1 = Constructor(144, 8, 4, 14)

print(result1)

result2 = Constructor(10000, 16, 6, 2)

print(result2)

6

def gen\_page(length, numbers=None):

if numbers is none:

numbers = [0] \* length

elif len(numbers) < length:

numbers.extend([0] \* (length - len(numbers)))

elif len(numbers) > length:

numbers = numbers[:length]

return numbers

result1 = generate\_list(5, [1, 2, 3])

print(result1)

result2 = generate\_list(3)

print(result2)

7

def custom\_filter(input\_list):

integers\_divisible\_by\_seven = [int(num) for num in input\_list if isinstance(num, (int, float)) and int(num) % 7 == 0]

total\_sum = sum(integers\_divisible\_by\_seven)

print(f"сумма: {total\_sum}\n")

return total\_sum > 83

print(custom\_filter([7, 10.5, 'txt', 14, 2, 56]))

8

def square(x, y):

print('\_' + '\_'.join(map(str, range(1, x + 1))) + '\_')

for i in range(1, y + 1):

print('|' + '|'.join(map(str, range(1, x + 1))) + '|')

print('¯' \* (x \* 2 + 1))

square(2, 3)