**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

###### **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

###### **КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ОТЧЁТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

«Функции высшего порядка»

Студентки 2 курса, ФИТ-211 группы

**Колесник Полины Олеговны**

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Руководитель:

Доцент Зимин А. И.

Работа защищена

« »

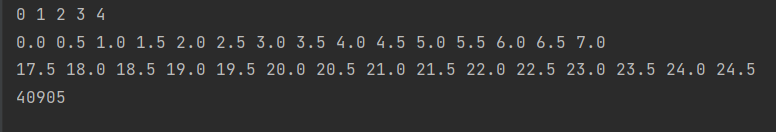
“ ” 2023 г.

Кемерово 2023 г.

**Основная часть.**

**Задание 1**

print(\*filter(lambda x: x < 5, range(10)))  
  
print(\*map(lambda x: x / 2, range(15)))  
  
print(\*map(lambda x: x / 2, filter(lambda x: x / 2 > 17, range(50))))  
  
print(sum(map(lambda x: x \*\* 2, filter(lambda x: not x % 9, range(10, 100)))))



**Задание 2**

def factorials(n):  
 result = 1  
 for i in range(1, n + 1):  
 result \*= i  
 yield result  
  
  
print(\*factorials(5))



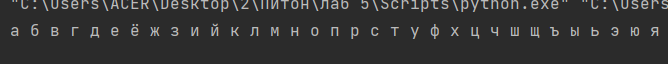
**Задание 3**

def square\_fibonacci(n):  
 result1, result2 = 0, 1  
 for i in range(n):  
 yield result2 \*\* 2  
 result1, result2 = result2, result1 + result2  
  
  
print(\*square\_fibonacci(7))



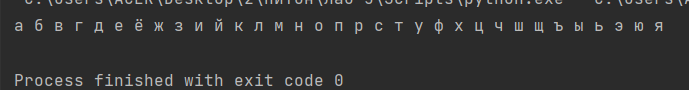
**Задание 4**

def alphabet():  
 for i in range(ord('а'), ord('е') + 1):  
 yield chr(i)  
 yield 'ё'  
 for i in range(ord('ж'), ord('я') + 1):  
 yield chr(i)  
  
  
print(\*alphabet())



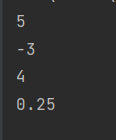
**Задание 5**

alphabet1 = [chr(i) for i in range(ord('а'), ord('ж'))]  
alphabet2 = [chr(i) for i in range(ord('ж'), ord('я') + 1)]  
  
print(\*alphabet1, 'ё', \*alphabet2)



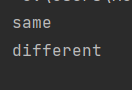
**Задание 6**

def arithmetic\_operation(operation):  
 if operation == '+':  
 return lambda a, b: a + b  
 elif operation == '-':  
 return lambda a, b: a - b  
 elif operation == '\*':  
 return lambda a, b: a \* b  
 elif operation == '/':  
 return lambda a, b: a / b  
 else:  
 return lambda a, b: 0  
  
  
operation = arithmetic\_operation('+')  
print(operation(1, 4))  
  
operation = arithmetic\_operation('-')  
print(operation(1, 4))  
  
operation = arithmetic\_operation('\*')  
print(operation(1, 4))  
  
operation = arithmetic\_operation('/')  
print(operation(1, 4))



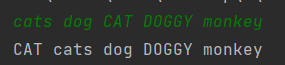
**Задание 7**

# 7  
def same\_by(characteristic, objects):  
 pred = objects[0] % 2  
 for i in objects:  
 if characteristic(i) == pred:  
 pred = characteristic(i)  
 continue  
 else:  
 return False  
 return True  
  
  
values = [0, 2, 10, 6]  
if same\_by(lambda x: x % 2, values):  
 print('same')  
else:  
 print('different')  
  
values = [1, 2, 3, 4]  
if same\_by(lambda x: x % 2, values):  
 print('same')  
else:  
 print('different')



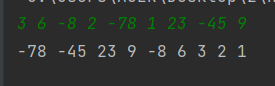
**Задание 10**

string = input().split()  
print(' '.join(sorted(string, key=lambda x: x.lower())))



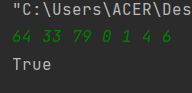
**Задание 11**

import math  
  
numbers = input().split()  
  
print(\*sorted(numbers, key=lambda x: abs(int(x)), reverse=True))



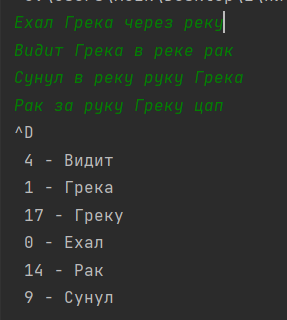
**Задание 13**

string = input().split()  
  
print(any(not all(map(int, x.split())) for x in string if x == '0'))



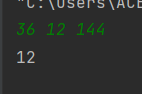
**Задание 14**

from sys import stdin  
  
lines = ''  
for line in stdin:  
 lines += line  
  
result = {}  
  
for i, val in enumerate(lines.split(), start=0):  
 if val[0].isupper() and val not in result:  
 result[val] = i  
  
for i, val in sorted(result.items()):  
 print(f' {val} - {i}')



**Задание 16**

import math  
import functools  
  
result = map(int, input().split())  
print(functools.reduce(math.gcd, result))



**Задание 17**

def check\_password(old\_f):  
 def new\_f(\*args, \*\*kwargs):  
 password = input()  
 if password != "123":  
 print("В доступе отказано")  
 return None  
 return old\_f(\*args, \*\*kwargs)  
 return new\_f()  
  
  
@check\_password  
def f1(\*args, \*\*kwargs):  
 print("Доступ разрешён")

