**1 Задание**

Задание:

Создание репозитория на сервисе Github через браузер.

1. Создайте репозиторий lab01 на сервисе Github через браузер.

2. Клонируйте репозиторий на компьютер при помощи GitKraken

3. Создайте проект в PyCharm в данном репозитории

4. Добавьте папку с виртуальным окружением venv в исключения

системы контроля версий

5. Создайте файл main.py в директории проекта с программой, которая

реализует алгоритм Евклида – алгоритм нахождения наибольшего

общего делителя (НОД) пары целых чисел.

6. Создайте коммит, отправьте его на сервер Github

7. Откройте в браузере репозиторий, убедитесь, что созданная программа

находится в репозитории.

Код: (Код для реализации алгоритма Евклида)

def NOD(a, b):  
 while a != 0 and b != 0:  
 if a > b:   
 a = a-b  
 else:   
 b = b-a  
 return max(a, b)  
  
def main():  
 print(NOD(30, 24))  
  
main()

Вывод:



**2 Задание**

Задание:

Создание репозитория на сервисе Github через программу

GitKraken.

1. Создайте репозиторий lab02 на сервисе Github через программу

GitKraken.

2. Создайте проект в PyCharm в данном репозитории

3. Добавьте папку с виртуальным окружением venv в исключения

системы контроля версий

4. Создайте файл main.py в директории проекта с программой, которая

реализует алгоритм решето Эратосфена – алгоритм нахождения

простых чисел до заданного натурального числа путем постепенного

отсеивания составных чисел.

5. Создайте коммит, отправьте его на сервер Github

6. Откройте в браузере репозиторий, убедитесь, что созданная программа

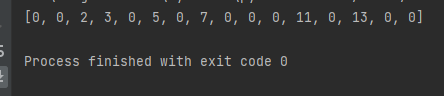
находится в репозитории. Убедитесь, что в облако не попала папка

venv.

Код: (Код для реализации алгоритма решето Эратосфена)

def simple(n):  
 sieve = list(range(n + 1))  
 sieve[1] = 0  
 for i in sieve:  
 if i > 1:  
 for j in range(2\*i, len(sieve), i):  
 sieve[j] = 0  
 return sieve  
  
def main():  
 print(simple(15))  
  
main()

Вывод:



**3 Задание**

Задание:

Основы работы с ветками

1. Создайте репозиторий lab03 на сервисе Github через программу

GitKraken.

2. Создайте проект в PyCharm в данном репозитории

3. Добавьте папку с виртуальным окружением venv в исключения

системы контроля версий

4. Создайте файл main.py в директории проекта с программой, которая

запрашивает с клавиатуры n чисел и сортирует их при помощи

алгоритма сортировки пузырьком по возрастанию и выводит их на

экран.

5. Создайте коммит, отправьте его на сервер Github

6. Откройте в браузере репозиторий, убедитесь, что созданная программа

находится в репозитории.

7. Создайте ветку от текущего коммита. Назовите ветку new-feature,

переключитесь на неё.

8. Измените программу так, чтобы она после чтения чисел запрашивала

направление сортировки (по возрастанию/по убыванию) и производила

сортировку соответствующим образом.

9. Создайте коммит, отправьте его на сервис Github.

10. Откройте в браузере репозиторий, убедитесь, что созданная программа

находится в репозитории.

11. При помощи программы GitKraken, слейте изменения из ветки new-

feature в основную ветку вашего репозитория.

12. Отправьте изменения на сервер GitHub.

13. Откройте в браузере репозиторий, убедитесь, что внесенные изменения

находится в репозитории.

Код: (Код для реализации алгоритма решето Эратосфена)

from random import randint  
  
def puzsort(arr):  
 for i in range(len(arr)-1):  
 for j in range(len(arr)-i-1):  
 if arr[j] > arr[j+1]:  
 buff = arr[j]  
 arr[j] = arr[j+1]  
 arr[j+1] = buff  
  
def main():  
 arr = []  
 for i in range(15):  
 arr.append(randint(1, 99))  
 print(arr)  
 puzsort(arr)  
 print(arr)  
  
main()

Вывод:



**4 Задание**

Задание:

Задание №4б.

1. Откройте на сайте Github проект коллеги lab04 и сделайте fork.

2. Выгрузите на компьютер созданный репозиторий.

3. Добавьте программу для тестирования модуля в файл test.py.

4. Создайте коммит и отправьте его на сервис Github.

5. Откройте в браузере репозиторий, убедитесь, что созданная программа

находится в репозитории.

6. Создайте запрос на слияние в исходный репозиторий.

7. Убедитесь, что слияние произошло успешно, обновленная программа

находится в исходном репозитории.

Код:

main.py: (Код для реализации для запуска программы)

import test  
from lib import lib  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 res = lib([[1, 2, 3], [3, 4, 5], [5, 6, 7], [7, 8, 9], [1, 2, 3]])  
 print(res)  
 print(test.test())

lib.py (функция, которая принимает на вход N списков и возвращает количество одинаковых элементов в них.)

def lib(data):  
 res = set()  
 for i in data:  
 for j in data:  
 if i is not j:  
 res |= set(i) & set(j)  
 return res

test.py (Код для тестирования функции lib)

import lib  
  
  
def test():  
 data = [[1, 2, 3], [3, 4, 5], [5, 6, 7], [7, 8, 9], [1, 2, 3]]  
 res = lib.lib(data)  
 check = False  
 expected = {1, 2, 3, 5, 7}  
 if expected == res:  
 check = True  
 return check

Вывод:

