МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №6 по дисциплине

«Разработка программных систем»

Выполнил студент группы ИВТ-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Седов М. Д./

Проверил доцент кафедры ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Долженкова М. Л./

Киров 2020

1 Задание

В ходе выполнения лабораторной работы необходимо решить следующие задачи:

* Установить на компьютер интерпретатор языка Python.
* Установить на компьютер подходящую среду разработки.
* Настроить рабочее окружение.
* Реализовать в отдельном модуле группу методов.
* Продемонстрировать работу модуля на подготовленном сценарии.

Модуль должен иметь четыре метода:

1) def get\_random\_prime() - возвращает случайное число из диапазона [2; 10^9]

2) def get\_random\_array\_prime() - возвращает массив простых чисел размерности length

3) def get\_random\_prime(m, n) - возвращает случайное простое число из диапазона [m; n]

4) def is\_relative\_primes(numbers) - возвращает true, если список numbers содержит взаимно-простые числа

5) def is\_pairwise\_primes(numbers) - возвращает true, если список numbers содержит попарно-простые числа

6) def get\_next(number) - возвращает первое простое число, следующее за number

7) def get\_prev(number) - возвращает первое простое число, предшествующее числу number

8) def is\_prime(number) - возвращает true, если number является простым числом

2 Листинг программы

Листинг программы приведен в приложении А.

1. Вывод

В ходе выполнения лабораторной работы был установлен интерпретатор Python версии 3.7.4, редактор сода VSCode; изучены основные конструкции языка Python, написан в функциональном стиле модуль для генерации массива простых чисел, а также программа для демонстрации его работы; изучен формат строк документации Python.

Приложение А

(обязательное) Листинг программы

**prime\_numbers.py**

MAX\_NUMBER = 999999937

def \_is\_prime(number):

    if number > 1:

        for i in range(2, number):

            if (number % i) == 0:

                return False

                break

        else:

            return True

    else:

        return False

def is\_prime(number):

    return number == 2 or \_is\_prime(number)

def get\_next(number, max\_value):

    lrec = lambda n: -1 if n == 10 \*\* 9 or n > max\_value else n if is\_prime(n) else lrec(n + 1)

    return lrec(number)

def get\_prev(number, min\_value):

    lrec = lambda n: -1 if n == 1 or n < min\_value else n if is\_prime(n) else lrec(n - 1)

    return lrec(number)

def get\_random\_prime\_number(m, n):

    from random import randint

    number = randint(m, n)

    if is\_prime(number):

        return number

    else:

        u = get\_next(number, n)

        if u == -1:

            v = get\_prev(number, m)

            if v == -1:

                return -1

            else:

                return v

        else:

            return u

def get\_random\_prime\_array(length):

    return sorted(get\_random\_prime\_number(2, 1000000) for \_ in range(length))

def is\_relative\_primes(numbers):

    from math import gcd

    result = numbers[0]

    for v in numbers[1:]:

        result = gcd(result, v)

        return True if result == 1 else False

def is\_pairwise\_primes(numbers):

    from math import gcd

    for i in range(len(numbers) - 1):

        for j in range(i + 1, len(numbers)):

            if gcd(numbers[i], numbers[j]) != 1:

                return False

    return True

**main.py**

import prime\_numbers as pn

MIN\_VALUE = 1

MAX\_VALUE = 10\*\*9

def read\_int(string, min\_value, max\_value):

    try:

        value = int(string)

        if min\_value <= value <= max\_value:

            return value

        return None

    except ValueError:

        return None

def read\_int\_array(s, min\_value, max\_value):

    values = s.split(' ')

    numbers = []

    for i in values:

        v = read\_int(i, min\_value, max\_value)

        if v is None:

            return None

        else:

            numbers.append(v)

    return numbers

def menu():

    print("Possible options:")

    print("1. Get random prime number(from 2 to n)")

    print("2. Get random prime number(from m to n)")

    print("3. Get array of prime numbers")

    print("4. Is relative primes(numbers)")

    print("5. Is pairwise primes(numbers)")

    print("6. Get next prime following the number")

    print("7. Get previous prime following the number")

    print("8. Is prime(number)")

    print("9. Quit")

    print("Select item:", end=' ')

if \_\_name\_\_ == "\_\_main\_\_":

    while True:

        menu()

        s = input()

        op\_code = read\_int(s, MIN\_VALUE, MAX\_VALUE)

        if op\_code is None:

            print("Unknow command!\n\n")

            continue

        elif op\_code == 1:

            print(pn.get\_random\_prime\_number(2, MAX\_VALUE))

            input("Enter for continue...")

        elif op\_code == 2:

            s = input("Enter min, max values: ")

            l = read\_int\_array(s, MIN\_VALUE, MAX\_VALUE)

            if l is None or len(l) != 2 or l[0] > l[1]:

                print("Wrong numbers")

            else:

                m, n = l[0], l[1]

                print(pn.get\_random\_prime\_number(m, n))

            input("Enter for continue...")

        elif op\_code == 3:

            s = input("Enter length of array: ")

            value = read\_int(s, MIN\_VALUE, MAX\_VALUE)

            if value is None:

                print("Wrong length of array")

            else:

                array = pn.get\_random\_prime\_array(value)

                print(\*array, sep=' ')

            input("Enter for continue...")

        elif op\_code == 4:

            s = input("Enter array values: ")

            l = read\_int\_array(s, MIN\_VALUE, MAX\_VALUE)

            if l is None:

                print("Wrong array")

            else:

                print(pn.is\_relative\_primes(l))

            input("Enter for continue...")

        elif op\_code == 5:

            s = input("Enter array values: ")

            l = read\_int\_array(s, MIN\_VALUE, MAX\_VALUE)

            if l is None:

                print("Wrong array")

            else:

                print(pn.is\_pairwise\_primes(l))

            input("Enter for continue...")

        elif op\_code == 6:

            s = input("Enter value: ")

            value = read\_int(s, MIN\_VALUE, MAX\_VALUE)

            if value is None:

                print("Wrong number")

            else:

                print(pn.get\_next(value, MAX\_VALUE))

            input("Enter for continue...")

        elif op\_code == 7:

            s = input("Enter value: ")

            value = read\_int(s, MIN\_VALUE, MAX\_VALUE)

            if value is None:

                print("Wrong number")

            else:

                print(pn.get\_prev(value, MIN\_VALUE))

            input("Enter for continue...")

        elif op\_code == 8:

            s = input()

            value = read\_int(s, MIN\_VALUE, MAX\_VALUE)

            if value is None:

                print("Value must be in range(1, 10^9)")

            else:

                print(pn.is\_prime(value))

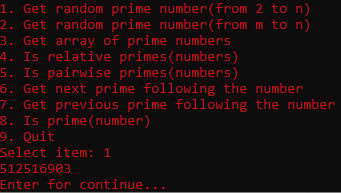
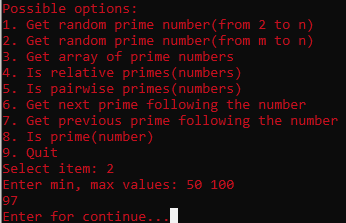
            input("Enter for continue...")

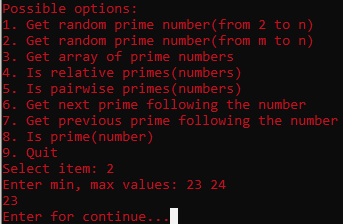
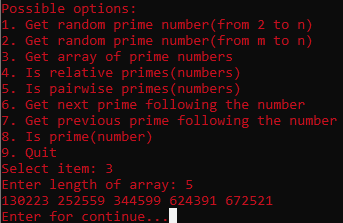
        elif op\_code == 9:

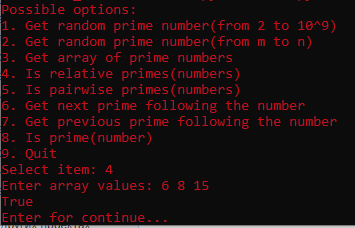
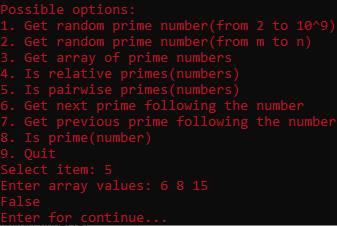
            break

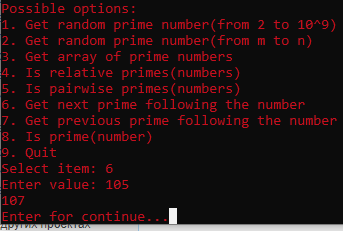
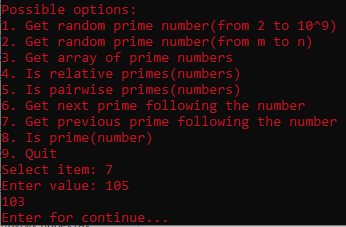
Приложение Б

Экранные формы

** **

**** ****

**** ****

**** ****