ФГБОУ ВО «Сочинский государственный университет»

Факультет Инновационных, Инженерных и Цифровых Технологий

Кафедра информационных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №2

*«»*

по дисциплине **«Алгоритмизация и программирование»**

***Выполнил:***

студент 1 курса гр.24-ПИЦ

Нерсесян С. В.

«\_\_»\_\_\_\_ \_\_\_\_2025г

***Проверил:***

проф. каф. инф. тех., д.т.н.

Попов Дмитрий Иванович

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_202-г

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отзыв, замечания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задача 1

Исходный код:

def fibSequence(n):

    assert n > 0

    series = [1]

    while len(series) < n:

        if len(series) == 1:

            series.append(1)

        else:

            series.append(series[-1] + series[-2])

    for i in range(len(series)):

        series[i] = str(series[i])

    return(', '.join(series))

print(fibSequence(int(input('Сколько чисел? '))))

Изменённый код:

def fibSequence(n):

    """Генерация последовательности Фибоначчи с n числами, используя итерационный подход"""

    assert n > 0

    series = [1]

    while len(series) < n:

        if len(series) == 1:

            series.append(1)

        else:

            series.append(series[-1] + series[-2])

    return series

def fibRecurse(n):

    """Рекурсивная генерация послесовательности Фибоначчи с n числами"""

    if n <= 2:

        return 1

    return fibRecurse(n-1) + fibRecurse(n-2)

def getFibSequence(n):

    """Получение последовательности Фибоначчи из n чисел, используя рекурсивную функцию."""

    return [fibRecurse(i+1) for i in range(n)]

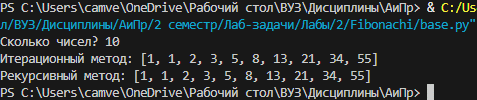
if \_\_name\_\_ == "\_\_main\_\_":

    n = int(input('Сколько чисел? '))

    print(f"Итерационный метод: {fibSequence(n)}")

    print(f"Рекурсивный метод: {getFibSequence(n)}")

Скриншоты:



Задача 2

Исходный код:

def isPrime(x):

    if x == 2:

        return True

    if x % 2 == 0:

        return False

    for i in range(3, int(x\*\*0.5)+1, 2):

        if x % i == 0:

            return False

    return True

def genPrime(currentPrime):

    newPrime = currentPrime + 1

    while True:

        if not isPrime(newPrime):

            newPrime += 1

        else:

            break

    return newPrime

currentPrime = 2

while True:

    answer = input('Показать следующее простое число? (Y/N) ')

    if answer.lower().startswith('y'):

        print(currentPrime)

        currentPrime = genPrime(currentPrime)

    else:

        break

Изменённый код:

def isPrime(x):

    """Проверка на простое число."""

    if x < 2:

        return False

    if x == 2:

        return True

    if x % 2 == 0:

        return False

    for i in range(3, int(x\*\*0.5)+1, 2):

        if x % i == 0:

            return False

    return True

def genPrime(currentPrime):

    """Генерация следующего простого числа."""

    newPrime = currentPrime + 1

    while True:

        if not isPrime(newPrime):

            newPrime += 1

        else:

            break

    return newPrime

def findNearestPrime(x):

    """Нахождение ближайшего простого числа."""

    if x < 2:

        return 2

    if isPrime(x):

        return x

    lower = x - 1

    higher = x + 1

    while True:

        if isPrime(lower):

            return lower

        elif isPrime(higher):

            return higher

        lower -= 1

        higher += 1

if \_\_name\_\_ == "\_\_main\_\_":

    while True:

        try:

            user\_input = input("Введите число для проверки на простоту (или 'q' для выхода): ")

            if user\_input.lower() == 'q':

                break

            number = int(user\_input)

            if isPrime(number):

                print(f"Число {number} является простым.")

            else:

                print(f"Число {number} не является простым.")

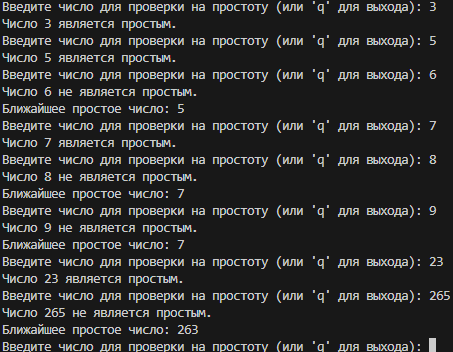
                nearest\_prime = findNearestPrime(number)

                print(f"Ближайшее простое число: {nearest\_prime}")

        except ValueError:

            print("Пожалуйста, введите целое число.")

Скриншоты:



Задача 3

Исходный код:

def calc(a, b, op):

    if op not in '+-/\*':

        return 'Пожалуйста, выберите тип операции: "+, -, \*, /"!'

    if op == '+':

        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a + b))

    if op == '-':

        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a - b))

    if op == '\*':

        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a \* b))

    if op == '/':

        return(str(a) + ' ' + op + ' ' + str(b) + ' = ' + str(a / b))

def main():

    a = int(input('Пожалуйста, введите первое число: '))

    b = int(input('Пожалуйста, введите второе число: '))

    op = input( 'Какой вид операции Вы желаете осуществить?\nВыберите между "+, -, \*, /" : ')

    print(calc(a, b, op))

if \_\_name\_\_ == '\_\_main\_\_':

    main()

Изменённый код:

import math

def calc(a, b, op):

    if op not in '+-/\*%^s':

        return 'Пожалуйста, выберите тип операции: "+, -, \*, /, %, ^, s"!'

    if op == '+':

        return f'{a} {op} {b} = {a + b}'

    if op == '-':

        return f'{a} {op} {b} = {a - b}'

    if op == '\*':

        return f'{a} {op} {b} = {a \* b}'

    if op == '/':

        if b == 0:

            return 'Деление на ноль запрещено!'

        return f'{a} {op} {b} = {a / b}'

    if op == '%':

        return f'{b}% от {a} = {(a \* b) / 100}'

    if op == '^':

        return f'{a}^{b} = {pow(a, b)}'

    if op == 's':

        if a < 0 or b < 0:

            return 'Квадратный корень из отрицательного числа невозможен!'

        return f'√{a} = {math.sqrt(a)} и √{b} = {math.sqrt(b)}'

def main():

    while True:

        try:

            print("\nВведите два числа:")

            a = float(input('Пожалуйста, введите первое число: '))

            b = float(input('Пожалуйста, введите второе число: '))

            print("\nВыберите операцию:")

            print("  + : Сложение")

            print("  - : Вычитание")

            print("  \* : Умножение")

            print("  / : Деление")

            print("  % : Вычисление процента")

            print("  ^ : Возведение в степень")

            print("  s : Извлечение квадратного корня из обоих чисел")

            op = input('\nКакой вид операции Вы желаете осуществить? ')

            result = calc(a, b, op)

            print(result)

            another = input('\nХотите выполнить еще одну операцию? (y/n): ')

            if another.lower() != 'y':

                break

        except ValueError:

            print('Ошибка: Пожалуйста, вводите только числовые значения!')

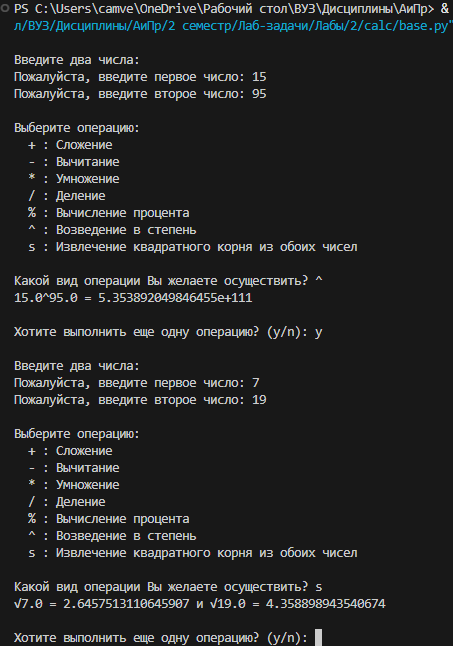
        except Exception as e:

            print(f'Произошла ошибка: {e}')

if \_\_name\_\_ == '\_\_main\_\_':

    main()

Скриншоты:



Задача 4

Исходный код:

import random

import os

import time

# Счет

score\_playera = 0

score\_bota = 0

# Начальное сообщение

all\_carts = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

print("Поиграем в 21? \nЕсли хотите играть нажмите Enter, если хотите выйти, то нажмите

Ctrl+C")

input()

# Цикл

while True:

    if score\_playera == 21:

        print("Больше карт не надо, у вас 21")

        print("Вы автоматически победили бота, так как у вас 21.")

        input("Нажмите Enter, чтобы закрыть окно."); break

    if score\_playera>21:

        print("Вы проиграли, так как набрали больше 21")

        print("Попытайте свою попытку в другой раз.")

        input("Нажмите Enter, чтобы закрыть окно."); break

    yes\_or\_no = input("Будете ли вы брать карту?\nВведите yes, если хотите брать карту

или введите no, если не берете карту.\n")

    os.system('cls')

    if yes\_or\_no == 'yes':

        os.system('cls')

        score\_carts = random.choice(all\_carts)

        print("Вы взяли карту выпало:", score\_carts)

        score\_playera += score\_carts

        print("Сейчас у вас ", score\_playera)

    if yes\_or\_no == 'no':

        print("У вас ", score\_playera, "очков.")

        print("Ход бота")

        time.sleep(3)

        os.system('cls')

        while True:

            if score\_bota<15:

                print("Бот берет карту")

                score\_carts = random.choice(all\_carts)

                print("Боту выпало", score\_carts, "очков.")

                score\_bota += score\_carts

                print("У бота ", score\_bota, "очков.")

                time.sleep(3)

                os.system('cls')

            if score\_bota>21:

                print("Бот проиграл.\nТак как у него", score\_bota, "очков, а у вас ",

score\_playera)

                input("Нажмите Enter, чтобы закрыть"); exit(0)

            if score\_bota>score\_playera:

                print("Бот победил.\nТак как у него", score\_bota, "очков, а у вас ",

score\_playera, "\nНе растраивайтесь. Попробуйте ещё раз.")

                input("Нажмите Enter, чтобы закрыть"); exit(0)

            if score\_bota == score\_playera:

                print("Вы набрали равное количество очков и у вас ничья")

                input("Нажмите Enter, чтобы закрыть"); exit(0)

Изменённый код:

import random

import os

import time

def main():

    score\_player = 0

    score\_bot = 0

    DIFFICULTY = 0.5

    all\_carts = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

    print("Поиграем в 21? \nЕсли хотите играть нажмите Enter, если хотите выйти, то нажмите Ctrl+C")

    input()

    while True:

        if score\_player == 21:

            print("Больше карт не надо, у вас 21")

            print("Вы автоматически победили бота, так как у вас 21.")

            input("Нажмите Enter, чтобы закрыть окно.")

            return

        if score\_player > 21:

            print("Вы проиграли, так как набрали больше 21")

            print("Попытайте свою попытку в другой раз.")

            input("Нажмите Enter, чтобы закрыть окно.")

            return

        yes\_or\_no = input("Будете ли вы брать карту?\nВведите yes, если хотите брать карту или введите no, если не берете карту.\n")

        os.system('cls')

        if yes\_or\_no.lower() == 'yes':

            score\_carts = random.choice(all\_carts)

            print("Вы взяли карту выпало:", score\_carts)

            score\_player += score\_carts

            print("Сейчас у вас ", score\_player)

        elif yes\_or\_no.lower() == 'no':

            print("У вас ", score\_player, "очков.")

            print("Ход бота")

            time.sleep(3)

            os.system('cls')

            while True:

                if score\_bot < 15:

                    if random.random() > DIFFICULTY:

                        print("Бот берет карту случайным образом")

                        score\_carts = random.choice(all\_carts)

                    else:

                        print("Бот использует стратегию для оптимальной игры")

                        ideal\_card = min((21 - score\_bot), max(2, 21 - score\_bot))

                        score\_carts = min(all\_carts, key=lambda x: abs(x - ideal\_card))

                    print("Боту выпало", score\_carts, "очков.")

                    score\_bot += score\_carts

                    print("У бота ", score\_bot, "очков.")

                    time.sleep(3)

                    os.system('cls')

                if score\_bot > 21:

                    print("Бот проиграл.\nТак как у него", score\_bot, "очков, а у вас ", score\_player)

                    input("Нажмите Enter, чтобы закрыть")

                    return

                if score\_bot > score\_player:

                    print("Бот победил.\nТак как у него", score\_bot, "очков, а у вас ", score\_player, "\nНе растраивайтесь. Попробуйте ещё раз.")

                    input("Нажмите Enter, чтобы закрыть")

                    return

                if score\_bot == score\_player:

                    print("Вы набрали равное количество очков и у вас ничья")

                    input("Нажмите Enter, чтобы закрыть")

                    return

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Скриншоты:

