

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ КАФЕДРА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №6
По дисциплине
«Информатика и программирование»

Студент _____ Д.В. Тимкин
гр. БИН-25-2

Ассистент _____ М.В. Водяницкий
преподавателя _____

Владивосток 2025

Задание

Выполнить задания на Python и оформить отчет по стандартам ВВГУ.

Задание 1.

Написать функцию, которая конвертирует время из одной величины в другую.

На вход подается:

- * число (величина времени) * исходная единица измерения * единица измерения, в которую нужно перевести

Функция должна вернуть конвертированное значение

Задание 2.

Пользователь делает вклад в банке в размере ‘*a*’ рублей сроком на ‘*n*’ лет

Процент по вкладу **зависит от суммы и срока**

Зависимость от суммы:

- * каждые 10 000 рублей увеличивают ставку на 0.3* но суммарное увеличение не может превышать 5* минимальный вклад - 30 000 рублей

Зависимость от срока:

- * первые 3 года - 3* от 4 до 6 лет - 5* более 6 лет - 2

Необходимо написать функцию, которая рассчитывает прибыль пользователя без учета первоначально вложенной суммы

Используется сложный процент: каждый год процент начисляется на текущую сумму вклада

На вход подаются: сумма вклада и количество лет. Результат: сумма прибыли (не весь вклад, а только заработанные проценты)

Задание 3.

Написать функцию для вывода всех простых чисел в заданном диапазоне. Нужно учитывать некорректные данные (например, начало больше конца или диапазон без простых чисел)

На вход подаются два числа: начало и конец диапазона (включительно). На выходе
 - список всех простых чисел или сообщение об ошибке
 (Формат вывода списка простых чисел может быть любым удобным: в строку
 через пробел, в несколько строк и т.п.)

Задание 4.

Реализовать функцию сложения двух матриц

При сложении двух матриц получается новая матрица того же размера, где каждый
 элемент - это сумма элементов с тем же индексом из двух исходных матриц

Ограничения:

* складывать можно только матрицы одинакового размера * размер матрицы должен
 быть строго больше 2 (например, 3×3 , 4×4 и т.д.) * при нарушении условий нужно вывести
 сообщение об ошибке

На вход подаются:

1. размер матрицы ‘n’ (для квадратной матрицы ‘ $n \times n$ ’) 2. элементы первой
 матрицы (по строкам, через пробел) 3. элементы второй матрицы в таком же формате

Результат - новая матрица (в том же формате), либо сообщение об ошибке

Пример (один из возможных вариантов формата):

Вход:

““txt 2 2 5 5 3 5 2 4 1 ““

Выход:

““txt 7 7 9 4 ““

Пример с ошибкой (слишком маленький размер, неправильный ввод и т.п.):

““txt 1 4 5 ““

Выход: ““txt

Error! ““

Задание 5.

Написать функцию, которая определяет, является ли строка палиндромом

Палиндром - это строка, которая читается одинаково слева направо и справа налево
 (обычно без учета пробелов, регистра и знаков препинания - эти правила нужно явно
 задать в своей реализации)

На вход подается строка. На выходе:

* ‘Да’, если это палиндром * ‘Нет’, если это не палиндром

Содержание

1 Выполнение работы	6
1.1 Задание 1	6
1.2 Задание 2	6
1.3 Задание 3	7
1.4 Задание 4	8
1.5 Задание 5	9

1 Выполнение работы

1.1 Задание 1

В данном задании была реализована программа для конвертации времени из одной единицы измерения в другую. Пользователь вводил числовое значение времени, исходную единицу измерения и целевую единицу. После проверки корректности введенных данных выполнялось преобразование и выводился результат.

```
value = float(input())
from_unit = input()
to_unit = input()

units = {"s": 1, "m": 60, "h": 3600}

if from_unit in units and to_unit in units:
    result = value * units[from_unit] / units[to_unit]
    print(result)
else:
    print("Error!")
```

Рисунок 1 — Листинг программы для задания 1

Пояснение работы программы:

- 1) Программа запрашивает числовое значение времени и сохраняет его в переменную `value`.
- 2) Пользователь вводит исходную и целевую единицы измерения времени.
- 3) Создается словарь `units`, содержащий коэффициенты перевода секунд, минут и часов.
- 4) Проверяется наличие введенных единиц в словаре `units`.
- 5) Если проверка пройдена, выполняется перевод величины времени.
- 6) Результат вычислений выводится на экран.
- 7) Если введена некорректная единица измерения, выводится сообщение об ошибке.

1.2 Задание 2

В данном задании была реализована программа для расчета прибыли по банковскому вкладу с использованием сложного процента. Процентная ставка зависела от суммы вклада и срока размещения средств. Результатом работы программы являлась сумма заработанных процентов.

```

amount = int(input())
years = int(input())

if amount < 30000 or years <= 0:
    print("Error!")
else:
    if years <= 3:
        rate = 0.03
    elif years <= 6:
        rate = 0.05
    else:
        rate = 0.02

    bonus = (amount // 10000) * 0.003
    if bonus > 0.05:
        bonus = 0.05

    rate += bonus

    total = amount
    for i in range(years):
        total += total * rate

    print(round(total - amount, 2))

```

Рисунок 2 — Листинг программы для задания 2

Пояснение работы программы:

- 1) Программа запрашивает сумму вклада и срок размещения в годах.
- 2) Проверяется корректность введенных данных.
- 3) В зависимости от срока вклада определяется базовая процентная ставка.
- 4) Рассчитывается дополнительная надбавка к ставке в зависимости от суммы вклада.
- 5) Надбавка ограничивается максимальным значением.
- 6) Итоговая процентная ставка формируется путем сложения базовой ставки и надбавки.
- 7) С помощью цикла рассчитывается итоговая сумма вклада по сложному проценту.
- 8) На экран выводится сумма прибыли без учета первоначального вклада.

1.3 Задание 3

В данном задании была разработана программа для поиска всех простых чисел в заданном диапазоне. Программа учитывала возможный некорректный ввод данных и выводила сообщение об ошибке при отсутствии простых чисел.

```

start = int(input())
end = int(input())

if start > end or end < 2:
    print("Error!")
else:
    found = False
    for num in range(max(2, start), end + 1):
        is_prime = True
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            print(num, end=" ")
            found = True

    if not found:
        print("Error!")

```

Рисунок 3 — Листинг программы задания 3

Пояснение работы программы:

- 1) Программа запрашивает начало и конец диапазона чисел.
- 2) Проверяется корректность введенного диапазона.
- 3) Запускается цикл перебора чисел в заданном диапазоне.
- 4) Для каждого числа выполняется проверка на простоту.
- 5) Проверка осуществляется делением на числа до квадратного корня.
- 6) Если число является простым, оно выводится на экран.
- 7) При отсутствии простых чисел выводится сообщение об ошибке.

1.4 Задание 4

В данном задании была реализована программа для сложения двух квадратных матриц одинакового размера. Перед выполнением операции проверялся размер матриц, и при нарушении условий выводилось сообщение об ошибке.

```

n = int(input())

if n <= 2:
    print("Error!")
else:
    a = []
    b = []

    for i in range(n):
        a.append(list(map(int, input().split())))

    for i in range(n):
        b.append(list(map(int, input().split())))

    for i in range(n):
        for j in range(n):
            print(a[i][j] + b[i][j], end=" ")
    print()

```

Рисунок 4 — Листинг программы задания 4

Пояснение работы программы:

- 1) Программа запрашивает размер квадратной матрицы n.
- 2) Проверяется условие, что размер матрицы больше двух.
- 3) Вводятся элементы первой матрицы построчно.
- 4) Вводятся элементы второй матрицы аналогичным образом.
- 5) Запускаются вложенные циклы для сложения соответствующих элементов.
- 6) Результаты сложения выводятся в виде новой матрицы.
- 7) При некорректном размере матрицы выводится сообщение об ошибке.

1.5 Задание 5

В данном задании была создана программа для определения, является ли введенная строка палиндромом. При проверке не учитывались регистр символов и небуквенно-цифровые знаки.

Результат проверки выводился на экран.

```
text = input()

clean = ""
for c in text:
    if c.isalnum():
        clean += c.lower()

if clean == clean[::-1]:
    print("Да")
else:
    print("Нет")
```

Рисунок 5 — Листинг программы задания 5

Пояснение работы программы:

- 1) Программа запрашивает ввод строки.
- 2) Создается пустая строка для хранения обработанного текста.
- 3) Выполняется перебор всех символов исходной строки.
- 4) В строку добавляются только буквенно-цифровые символы в нижнем регистре.
- 5) Полученная строка сравнивается с перевернутым вариантом.
- 6) Если строки совпадают, выводится сообщение «Да».
- 7) Если строки не совпадают, выводится сообщение «Нет».