

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 №4
по дисциплине
«Информатика и программирование»

Студент
гр. БИС-25-3

Л.С.Лавриненко

Ассистент
преподавателя

М. В. Водяницкий

Задание

Выполнить задание 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 и единица измерения, в которую нужно перевести. Функция должна вернуть конвертированное значение.

Задание 2. Рассчитать прибыль по вкладу с учетом сложного процента и зависимостью ставки от суммы и срока вклада. Минимальный вклад - 30 000 рублей. Каждые 10 000 рублей увеличивают ставку на 0.3%, но суммарное увеличение не может превышать 5%. Ставка также зависит от срока: первые 3 года - 3%, от 4 до 6 лет - 5%, более 6 лет - 2%.

Задание 3. Написать функцию для вывода всех простых чисел в заданном диапазоне с обработкой некорректных данных. На вход подаются два числа: начало и конец диапазона (включительно). На выходе - список всех простых чисел или сообщение об ошибке.

Задание 4. Реализовать функцию сложения двух квадратных матриц одинакового размера больше 2×2 . При нарушении условий нужно вывести сообщение об ошибке.

Задание 5. Написать функцию, которая определяет, является ли строка палиндромом (без учета пробелов, регистра и знаков препинания). На вход подается строка. На выходе: "Да", если это палиндром, или "Нет", если это не палиндром.

Содержание

1	Выполнение работы	5
1.1	Задание 1	5
1.2	Задание 2	6
1.3	Задание 3	8
1.4	Задание 4	9
1.5	Задание 5	11

1 Выполнение работы

1.1 Задание 1

В этом задании необходимо написать функцию для конвертации времени из одной единицы измерения в другую. На вход подаются три значения: величина времени (число), исходная единица измерения (часы "h", минуты "m" или секунды "s") и целевая единица измерения. Функция выполняет преобразование через базовую единицу — секунды. Сначала исходное значение переводится в секунды, затем конвертируется в требуемую единицу измерения. Результат выводится с точностью до двух знаков после запятой.

```
def task1():
    data = input().split()
    num = float(data[0])
    from_u = data[1]
    to_u = data[2]

    # Конвертируем всё в секунды
    if from_u == "h":
        num = num * 3600
    elif from_u == "m":
        num = num * 60

    # Конвертируем из секунд в нужную единицу
    if to_u == "h":
        num = num / 3600
    elif to_u == "m":
        num = num / 60

    # Форматируем вывод
    if num == int(num):
        print(f"{int(num)}{to_u}")
    else:
        print(f"{num:.2f}{to_u}")
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Пользователь вводит данные в формате: число, исходная единица измерения и целевая единица
- 2) Значение конвертируется в секунды (базовая единица).
- 3) Если исходная единица - часы, значение умножается на 3600.
- 4) Если исходная единица - минуты, значение умножается на 60.
- 5) Затем значение конвертируется из секунд в целевую единицу.
- 6) Если целевая единица - часы, значение делится на 3600.
- 7) Если целевая единица - минуты, значение делится на 60
- 8) Результат выводится с округлением до двух знаков после запятой

1.2 Задание 2

В этом задании требуется рассчитать прибыль по банковскому вкладу с учётом сложного процента. Ставка по вкладу зависит от двух параметров: суммы вклада и срока размещения. От суммы: каждые 10 000 рублей увеличивают ставку на 0,3%, но не более чем на 5% суммарно. От срока: 3% для вкладов до 3 лет включительно, 5% — от 4 до 6 лет, 2% — для сроков более 6 лет. Минимальная сумма вклада составляет 30 000 рублей. Программа вычисляет итоговую сумму с начисленными сложными процентами и выводит только прибыль (разницу между итоговой и начальной суммой).

```

def task2():
    a, n = map(int, input().split())

    if a < 30000:
        print("Ошибка")
        return

    # Бонус за сумму: каждые 10000 руб = +0.3%, максимум +5%
    bonus = (a // 10000) * 0.3
    if bonus > 5:
        bonus = 5

    # Базовая ставка по сроку
    if n <= 3:
        rate = 3.0
    elif n <= 6: # 4-6 лет
        rate = 5.0
    else: # более 6 лет
        rate = 2.0

    # Итоговая ставка
    rate = rate + bonus

    # Считаем сложный процент
    money = float(a)
    for _ in range(n):
        money = money + money * (rate / 100)

    # Прибыль
    profit = money - a
    print(f"{profit:.2f}")

```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Пользователь вводит сумму вклада и срок в годах.
- 2) Проверяется минимальная сумма вклада (30 000 рублей).
- 3) Рассчитывается бонус за сумму: каждые 10 000 рублей дают +0.3%, но не более +5%.

- 4) Определяется базовая ставка по сроку: 3% для 1-3 лет, 5% для 4-6 лет, 2% для более 6 лет.
- 5) Суммируется базовая ставка и бонус .
- 6) Рассчитывается сложный процент: каждый год сумма увеличивается на процент от текущей суммы.
- 7) Вычисляется прибыль как разница между итоговой и начальной суммой
- 8) Результат выводится с округлением до двух знаков.

1.3 Задание 3

В этом задании необходимо найти все простые числа в заданном пользователем диапазоне. Диапазон включает начало и конец. Программа проверяет корректность ввода: начало не должно превышать конец, а также проверяет наличие возможных простых чисел в диапазоне. Для каждого числа в диапазоне выполняется проверка на простоту: число считается простым, если оно больше 1 и делится без остатка только на 1 и на само себя. Если простые числа найдены, они выводятся списком; если нет или введены некорректные данные, выводится сообщение об ошибке.

```
def task3():
    start, end = map(int, input().split())

    if start > end:
        print("Error!")
        return

    primes = []
    for num in range(start, end + 1):
        if num < 2:
            continue
        prime = True
        for i in range(2, int(num**0.5) + 1):
            if num % i == 0:
                prime = False
                break
        if prime:
            primes.append(str(num))

    if not primes:
        print("Error!")
    else:
        print(" ".join(primes))
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Пользователь вводит начало и конец диапазона.
- 2) Проверяется корректность диапазона (начало \leq конец).
- 3) Для каждого числа в диапазоне проверяется, является ли оно простым.
- 4) Простое число проверяется делением на все числа от 2 до квадратного корня из этого числа.
- 5) Если число делится без остатка на какой-либо делитель, оно не является простым.
- 6) Все найденные простые числа добавляются в список .
- 7) Если список пуст или диапазон некорректен, выводится сообщение об ошибке.
- 8) В противном случае выводится список простых чисел

1.4 Задание 4

В этом задании требуется реализовать сложение двух квадратных матриц одинакового размера. Матрицы должны быть размером больше 2×2 (например, 3×3 , 4×4 и т.д.). Пользователь вводит размер матрицы, затем поэлементно вводит значения для первой и второй матриц. Программа проверяет корректность ввода: соответствие размеров и возможность сложения. Если условия выполняются, создаётся новая матрица, каждый элемент которой равен сумме соответствующих элементов исходных матриц. Результат выводится в виде матрицы. При нарушении условий выводится сообщение об ошибке.


```

def task4():
    n = int(input())

    if n <= 2:
        print("Error!")
        return

    matrix1 = []
    for _ in range(n):
        row = list(map(int, input().split()))
        matrix1.append(row)

    matrix2 = []
    for _ in range(n):
        row = list(map(int, input().split()))
        matrix2.append(row)

    result = []
    for i in range(n):
        row = []
        for j in range(n):
            row.append(matrix1[i][j] + matrix2[i][j])
        result.append(row)

    for row in result:
        print(" ".join(map(str, row)))

```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Пользователь вводит размер квадратной матрицы.
- 2) Проверяется, что размер больше 2.
- 3) Считываются элементы первой матрицы построчно.
- 4) Считываются элементы второй матрицы построчно.
- 5) Проверяется соответствие количества элементов в строках.
- 6) Создается новая матрица того же размера.
- 7) Каждый элемент новой матрицы вычисляется как сумма соответствующих элементов исходных матриц.
- 8) Результат выводится в виде матрицы.

1.5 Задание 5

В этом задании необходимо определить, является ли введённая строка палиндромом. Палиндром — это строка, которая читается одинаково слева направо и справа налево. При проверке игнорируются регистр букв, пробелы и знаки препинания. Программа очищает строку, оставляя только буквы, приводит их к нижнему регистру, а затем сравнивает полученную строку с её обратной версией. Если строки совпадают, выводится "Да", в противном случае — "Нет"

```
def task5():
    text = input().lower()

    cleaned = ""
    for char in text:
        if char.isalpha():
            cleaned += char

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

- 1) Пользователь вводит строку
- 2) Строка приводится к нижнему регистру
- 3) Удаляются все символы, кроме букв (русских и английских)
- 4) Полученная строка сравнивается с той же строкой, записанной в обратном порядке
- 5) Если строки идентичны, выводится "Да"
- 6) Если строки различаются, выводится "Нет"

