

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
гр. БИН-25-2 _____ А.В. Воронин
Ассистент
преподавателя _____ М.В. Водяницкий

Задание

Выполнить задания и оформить отчет по стандартам ВВГУ.

Задание 1. Написать функцию, которая конвертирует время из одной величины в другую. На вход подаются:

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

Функция должна вернуть конвертированное значение.

Пример:

Введите время для конвертации: 4 h m

Результат: 240m

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%
- 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.

Пример:

Ввод: 30000 3

Прибыль: 3648.67 руб.

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.

Пример:

Начало диапазона: 1

Конец диапазона: 10

2 3 5 7

Пример:

Начало диапазона: 0

Конец диапазона: 1

Error!

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.

Задание 5. Написать функцию, которая определяет, является ли строка палиндромом.

Палиндром — это строка, которая читается одинаково слева направо и справа налево (без учета пробелов, регистра и знаков препинания).

На вход подается строка. На выходе:

- Да, если это палиндром
- Нет, если это не палиндром

Пример:

А роза упала на лапу Азора

Да

Пример:

Алфавитный порядок

Нет

Содержание

1 Выполнение работы	3
1.1 Задание 1	3
1.2 Задание 2	3
1.3 Задание 3	4
1.4 Задание 4	5
1.5 Задание 5	5

1 Выполнение работы

1.1 Задание 1

Реализована функция `time_convert()`, предназначенная для конвертации времени между различными единицами измерения: секунды (`s`), минуты (`m`), часы (`h`) и дни (`d`). Внутри функции определён словарь `time`, содержащий ключи-синонимы единиц времени и соответствующие значения в секундах. Пользователь вводит строку в формате «число исходная_единица целевая_единица». Программа разбивает ввод на три компонента, преобразует значение в вещественное число и приводит единицы к нижнему регистру. Если обе единицы распознаны, выполняется перевод по формуле:

$$\text{результат} = \frac{\text{значение} \cdot \text{коэффициент_исходной}}{\text{коэффициент_целевой}}.$$

Результат выводится как целое число (если возможно) или с двумя знаками после запятой. Обрабатываются ошибки формата и неизвестных единиц. На рисунке 1 представлен код программы.

```

1 def time_convert():
2
3     time = {
4         's': 1, 'sec': 1, 'second': 1,
5         'm': 60, 'min': 60, 'minute': 60,
6         'h': 3600, 'hr': 3600, 'hour': 3600,
7         'd': 86400, 'day': 86400
8     }
9
10    user_input = input("Введите время для конвертации: ")
11
12    try:
13        value, from_time, to_time = user_input.split()
14        value = float(value)
15        from_time = from_time.lower()
16        to_time = to_time.lower()
17
18        if from_time in time and to_time in time:
19            result = value * time[from_time] / time[to_time]
20            if result.is_integer():
21                print(f"Результат: {int(result)}{to_time}")
22            else:
23                print(f"Результат: {result:.2f}{to_time}")
24        else:
25            print("Ошибка: неизвестная единица измерения.")
26
27    except ValueError:
28        print("Ошибка: неверный формат ввода.")
29
30 time_convert()

```

Рисунок 1 – Листинг программы для задания 1

1.2 Задание 2

Реализована функция `calculate_profit(a, n)`, вычисляющая прибыль по банковскому вкладу с учётом сложного процента. Сначала проверяется, что сумма вклада не

менее 30 000 руб. Ставка по сроку определяется условной конструкцией: 3% (≤ 3 лет), 5% (4–6 лет), 2% (> 6 лет). Дополнительная ставка за сумму рассчитывается как $\min((a // 10000) * 0.003, 0.05)$. Итоговая прибыль вычисляется по формуле сложного процента:

$$\text{прибыль} = a \cdot (1 + r_{\text{срок}} + r_{\text{сумма}})^n - a.$$

Результат округляется до двух знаков. Обёртка `main_compact()` обеспечивает удобный ввод в цикле. На рисунке 2 представлен код программы.

```

1 def calculate_profit(a, n):
2     if a < 30000: return "Ошибка: минимальная сумма 30 000
3         руб."
4
5     term = 0.03 if n <= 3 else 0.05 if n <= 6 else 0.02
6     sum = min((a // 10000) * 0.003, 0.05)
7
8     profit = a * (1 + term + sum) ** n - a
9     return f"Прибыль: {round(profit, 2):,.2f} руб."
10
11 def main_compact():
12     print("Калькулятор вклада сумма(, срок):")
13     while True:
14         try:
15             data = input("Ввод: ").split()
16             if len(data) != 2: continue
17
18             amount, years = float(data[0]), int(data[1])
19             result = calculate_profit(amount, years)
20             print(result)
21
22         except (ValueError, IndexError):
23             print("Ошибка ввода!")
24         except KeyboardInterrupt:
25             print("\Выход"); break
26 main_compact()

```

Рисунок 2 – Листинг программы для задания 2

1.3 Задание 3

Реализована функция `primes(a, b)`, возвращающая список всех простых чисел в диапазоне $[a, b]$. Перебор начинается с $\max(2, a)$, так как простые числа начинаются с 2. Для каждого числа проверяется делимость нацело на числа от 2 до \sqrt{num} . Если делителей нет, число добавляется в список. Если список пуст, возвращается строка "Error!". Ввод границ диапазона осуществляется с консоли. На рисунке 3 представлен код программы.

```

1 def primes(a,b):
2     primes_list = []
3     for num in range(max(2, a), b + 1):
4         if num > 1:
5             for i in range (2, int(num**0.5) + 1):
6                 if num % i == 0:
7                     break
8             else:
9                 primes_list.append(num)
10
11 return primes_list if primes_list else "Error!"
12
13 a = int(input("Начало диапазона:"))
14 b = int(input("Конец диапазона:"))
15 print(primes(a, b))
16
17 primes(a,b)

```

Рисунок 3 – Листинг программы для задания 3

1.4 Задание 4

Реализована функция `add_matrices()`, выполняющая сложение двух квадратных матриц размера $n \times n$. Сначала запрашивается размер n ; если $n \leq 2$, выводится ошибка. Затем пользователь вводит две матрицы построчно. Каждая строка преобразуется в список целых чисел. После ввода выполняется поэлементное сложение и вывод результата. Любая ошибка ввода (например, недостаточное количество чисел) перехватывается общим исключением, и выводится `Error!`. На рисунке 4 представлен код программы.

```

1 def add_matrices():
2     try:
3         n = int(input("Введите размер матрицы: "))
4         if n <= 2:
5             print("Error!")
6             return
7
8         print("Введите первую матрицу:")
9         matrix1 = [list(map(int, input().split())) for _ in
range(n)]
10
11        print("Введите вторую матрицу:")
12        matrix2 = [list(map(int, input().split())) for _ in
range(n)]
13
14        print("Результат:")
15        for i in range(n):
16            print(' '.join(str(matrix1[i][j] + matrix2[i][j])
)) for j in range(n)))
17
18    except:
19        print("Error!")
20
21 add_matrices()

```

Рисунок 4 – Листинг программы для задания 4

1.5 Задание 5

Реализована проверка строки на палиндром. Стока очищается от всех не-буквенно-цифровых символов с помощью генератора `c.lower() for c in s if c.isalnum()`, после чего сравнивается с её обратным порядком (`cleaned[::-1]`). Если строки совпадают, выводится Да, иначе — Нет. Программа корректно обрабатывает смешанный регистр, пробелы и пунктуацию. На рисунке 5 представлен код программы.

```
1 s = input().strip()
2 cleaned = ''.join(c.lower() for c in s if c.isalnum())
3 print("Да" if cleaned == cleaned[::-1] else "Нет")
```

Рисунок 5 – Листинг программы для задания 5

Таким образом, все пять заданий лабораторной работы №6 выполнены в полном объёме. Программы корректно обрабатывают ошибки ввода, соответствуют постановке задач и протестированы на примерах из условия. Отчёт оформлен в соответствии с требованиями СТО ВВГУ.