

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 основы
программирования»

Студент
гр. БИС-25-3

Ассистент

преподавателя

_____ Д. А. Пестерев

_____ М. В. Водяницкий

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.

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:

- размер матрицы n (для квадратной матрицы $n \times n$)
- элементы первой матрицы (по строкам, через пробел)
- 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 (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.

Оглавление

1	Выполнение работы	5
1.1	Задание 1.....	5
1.2	Задание 2.....	6
1.3	Задание 3.....	7
1.4	Задание 4.....	8
1.5	Задание 5.....	10

1 Выполнение работы

1.1 Задание 1

В задании необходимо было написать программу, которая конвертирует время из одной величины в другую с помощью функции. На рисунке 1 представлен код программы.

```

1  # Задание 1
2  time = int(input('введите время: '))
3  t1 = input('введите исходную единицу измерения: ')
4  t2 = input('введите, в какую единицу измерения необходимо перевести: ')
5
6  def time_transfer(num, original, new): 1 usage
7      if original == 'h':
8          hour = num
9      elif original == 'm':
10         hour = num / 60
11     elif original == 's':
12         hour = num / 3600
13
14     if new == 'h':
15         return hour
16     elif new == 'm':
17         return hour * 60
18     elif new == 's':
19         return hour * 3600
20
21     print(f'результат: {time_transfer(time, t1, t2)}{t2}')
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

1) С помощью функции `input()` пользователь вводит через консоль количество единиц времени, далее вводит исходную единицу измерения, а после вводит единицу измерения, в которую необходимо перевести.

2) Задаём функцию `time_transfer` для превода времени, в функции задаём три значения: кол-во времени, исходную единицу, а также ту, в которую переводим.

3) Далее с помощью проверки `if` и доп условий `elif` проверяем исходную единицу и в соответствии с этим выполняем перевод нашего времени в часы посредством деления

4) Затем также с помощью условия `if` и доп условий проверяем в какую единицу необходимо перевести наше время и в соответствии с этим возвращаем результат функции, а именно переменную `hour`, умноженную на соответствующее число.

5) На экран выводим значение возвращённое функцией, а также рядом единицу в которую перевели время

1.2 Задание 2

В данном задании необходимо было написать программу, которая рассчитывает прибыль пользователя без учета первоначально вложенной суммы с помощью функции. На рисунке 2 представлен код программы.

```

1  # Задание 2
2  deposit = input('Введите сумму вклада и срок хранения через пробел: ')
3  sum, years = deposit.split(' ')
4  sum = int(sum)
5  years = int(years)
6
7  def money_back(sum, years): 1 usage
8      p = 0
9      money = 0
10     p_m = 0
11     if sum < 30000:
12         return 'Минимальная сумма вклада: 30 000'
13     for i in range(1, years + 1):
14         ost = sum // 10000
15         p = ost * 0.3
16         ost = 0
17         if p > 5:
18             p = 5
19         if i <= 3:
20             p += 3
21         elif i > 3 and i <= 6:
22             p += 5
23         elif i > 6:
24             p += 2
25         money += (sum * p)/100
26         p_m += money
27         sum += money
28         money = 0
29     return p_m
30
31 print(f'результат: {round(money_back(sum, years), 2)}')
```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

1) Через input запрашиваем сумму вклада и срок хранения через пробел. Через split разделяем на две переменные sum и years, а также переводим их в int.

2) Задаём функцию money_back. Создаём переменную p для записи процента, далее money для суммы которая начисляется по процентам, а также p_m для записи всех сумм money, то есть начислений по процентам за все года. Также проверяем чтобы минимальная сумма вклада была не меньше 30 000.

3) С помощью цикла for перебираем диапазон от 1 до количества лет, вычисляем количество деления вклада на 10 000, после этого вычисляем процент умножая кол-во на 0.3. Устанавливаем лимит для процентов до 5.

4) Далее проверяем по условиям какой идёт год, то есть чему равно i и в соответствии с этим добавляем дополнительные проценты, далее высчитываем сумму которая начислилась в этом году по процентам и прибавляем её к сумме всех сумм по процентам то есть к p_m . Также прибавляем сумму по процентам к первоначальному вкладу и обнуляем сумму по процентам для расчёта следующих годов. По завершению функции возвращаем p_m , то есть всю накопленную сумму по процентам за все года.

5) На экран выводится значение возвращённое функцией, округлённое до двух.

1.3 Задание 3

В данном задании необходимо было написать программу которая будет выводить все простые числа в указанном диапазоне с помощью функции. На рисунке 3 представлен код программы.

```

1  # Задание 3
2  d = input('введите диапазон, числа через пробел: ')
3  d1, d2 = d.split(' ')
4  d1 = int(d1)
5  d2 = int(d2)
6
7  def numbers(a, b): 1 usage
8      d = 0
9      l = []
10     if a > b:
11         return 'начало диапазона не может быть больше конца'
12     else:
13         for i in range(a, b + 1):
14             n = i
15             for j in range(1, b + 1):
16                 if n % j == 0:
17                     d += 1
18             if d == 2:
19                 l.append(n)
20                 d = 0
21             else:
22                 d = 0
23         return l
24
25 print(f'результат: {numbers(d1, d2)}')
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

1) Через функцию `input` пользователь вводит в консоль два числа через пробел, это диапазон в котором мы будем искать простые числа. Далее через метод `split` мы разделяем введенную строку по пробелам на две переменные: Начало диапазона и конец.

2) Задаём функцию `numbers()` и в неё передаём две переменные: начало и конец диапазона. Внутри функции задаём переменную `d` для подсчёта делителей и создаём пустой список, в который мы будем записывать простые числа. Также сразу задаём проверку на то, что начало диапазона не может быть больше конца.

3) Если с введенными данными всё в порядке, начинаем искать простые числа: с помощью цикла `for` перебираем все числа от начала диапазона до конца включительно. Создаём переменную `n`, задаём ей значение `i`.

4) Следующим шагом в цикле для текущего элемента создаём ещё один цикл `for` в котором мы уже перебираем числа от 1 до конца диапазона, чтобы найти делители, для этого мы делим `n` на каждое перебираемое число, и если остаток от деления равен нулю, увеличиваем переменную `d` на +1 для подсчёта делителей.

5) Если в конце цикла число делителей равно двум, то перебираемое число `n` простое число и мы добавляем его в список простых чисел, после обнуляем переменную `d` и начинаем перебирать следующее число `n`. По окончании всего перебора завершаем функцию и возвращаем заполненный список простых чисел.

6) На экран выводится значение возвращённое функцией.

1.4 Задание 4

В этом задании необходимо было написать программу, которая будет производить сложение двух матриц с помощью функций. На рисунке 4 представлен код программы.


```

1  # Задание 4
2  n = int(input('Введите размерность матрицы начиная от 3: '))
3  if n < 3:
4      print("Размерность матрицы должна быть не меньше 3-х")
5      exit()
6
7  matrix_result = []
8  matrix1 = []
9  matrix2 = []
10
11 def matrix(m): 2 usages
12     for i in range(n):
13         row = input(f'Введите строку размерности {n}x{n}: ').split()
14         if len(row) != n:
15             print('некорректный ввод')
16             exit()
17         m.append(list(map(int, row)))
18     return m
19
20 def matrix_sum(m1, m2, m_r): 1 usage
21     print('матрица 1: ')
22     m1 = matrix(m1)
23     print('матрица 2: ')
24     m2 = matrix(m2)
25
26     for i in range(n):
27         row = []
28         for j in range(n):
29             row.append(m1[i][j] + m2[i][j])
30         m_r.append(row)
31     return m_r
32
33 matrix_result = matrix_sum(matrix1, matrix2, matrix_result)
34 print('результат: ')
35 for row in matrix_result:
36     print(*row)

```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

1) С помощью функции `input` запрашиваем размерность матрицы с типом `int` и сразу делаем проверку на то, что размерность должна быть не меньше 3. Также создаём пустные списки для двух матриц над которыми будет производиться операция сложения и для матрицы результата.

2) Создаём функцию `matrix` для ввода наших матриц через консоль для этого создаём цикл и перебираем в диапазоне размерности матрицы, далее запрашиваем в консоль написать первую строку матрицы через пробел в соответствии с размерностью, сразу

разделяем все элементы и делаем проверку на то, что в строке нужное количество элементов, если кол-во элементов не равно размерности – выдаём ошибку. После проверки добавляем строку в матрицу перед этим с помощью `map` применяем `int` к каждому элементу в строке, по завершению цикла возвращаем матрицу.

3) Задаём следующую функцию `matrix_sum` для сложения матриц. В этой функции сразу же применяем функцию для ввода матрицы для двух матриц. Далее производим само сложение: задаём цикл `for` с диапазоном размерности, в нём создаём пустой список для строк, после создаём вложенный цикл для перебора колонок в строке: записываем в строку элемент, который является сложением элемента `[i][j]` матрицы 1 и элемента `[i][j]` матрицы 2. По завершению строки добавляем её в матрицу результата, после завершения функции возвращаем матрицу результата.

4) В `matrix_result` записываем значение возвращённое функцией сложения матриц. После на экран выводится матрица результата с помощью оператора распаковки списка.

1.5 Задание 5

В данном задании необходимо было написать программу, которая будет определять является ли, введённая пользователем с консоли, строка палиндромом с помощью функции. На рисунке 5 представлен код программы.

```

1  # Задание 5
2  w = input('Введите предложение: ')
3
4  def palindrome(word): 1 usage
5      p = []
6      for i in word:
7          if i.isalpha():
8              i = i.lower()
9              p.append(i)
10
11     p_new = p[::-1]
12     if p == p_new:
13         return 'палиндром'
14     else:
15         return 'не палиндром'
16
17     print(f'Это {palindrome(w)}')
```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Через функцию `input` запрашиваем на вход в консоль предложение.
- 2) Задаём функцию `palindrome`, в которую передаём значение `word` – это будет наше предложение. Создаём пустой список для перезаписи нашего предложения для удобной проверки.
- 3) Начинаем цикл `for`, перебираем все элементы в предложении. Проверяем с помощью `if`, что элемент является именно буквой, а не числом или символом. После приравниваем нашу букву к нижнему регистру и добавляем её в список.
- 4) Далее после заполнения списка создаём ещё один список `r_new`, в него записываем наш отсортированный список, перевёрнутый с помощью срезов. После просто проверяем равны ли два списка или нет и возвращаем соответствующие значения.
- 5) На экран выводим значение возвращённое функцией `palindrome`.