

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. БИН-25-3	_____	В.Е. Соловьева
Ассистент		
преподавателя	_____	М.В. Водяницкий

## 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:

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

Функция должна вернуть конвертированное значение

Примеры (формат ввода/вывода можно выбрать свой, если нет строгих требований):

Таблица 1

Вход	Выход
4h m	240m
30m h	0.5h
12s h	0.03h

**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:

- 1) каждые 10 000 рублей увеличивают ставку на 0.3%
- 2) но суммарное увеличение не может превышать 5%
- 3) минимальный вклад - 30 000 рублей

Зависимость от срока:

- 1) первые 3 года - 3%
- 2) от 4 до 6 лет - 5%
- 3) 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)

Примеры:

Таблица 2

Вход	Выход
30000 3	3648.67
100000 5	38920.10
200000 8	183925.42

**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

Примеры:

Таблица 3

Вход	Выход
1 10	2 3 5 7
15 120	17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113
0 1	Error!

(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:

- 1) складывать можно только матрицы одинакового размера
- 2) размер матрицы должен быть строго больше 2 (например, 3×3, 4×4 и т.д.)
- 3) 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:

```
2
2 5
5 3
5 2
4 1
```

Выход:

```
7 7
9 4
```

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

```
1
4
5
```

Выход:

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:

- 1) Да, если это палиндром
- 2) Нет, если это не палиндром

Примеры:

Таблица 4

Вход	Выход
А роза упала на лапу Азора	Да
Borrow or rob	Да
Алфавитный порядок	Нет

**Оформление отчета.** Отчет оформляется строго по СТО - скачать требования можно с сайта ВВГУ, либо отсюда

Не забудьте добавить страницу "Задание" с копией содержимого этого файла (с правильным оформлением списков и т.д.)

В отчете должно быть объяснено как работает ваша программа (каждое отдельное задание)

## Содержание

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

## 1 Выполнение работы

### 1.1 Задание 1

В данном задании необходимо написать функцию, которая конвертирует время из одной величины в другую. На рисунке 1 представлен код полученной программы.

```

1 initial = input('Введите величину времени, исходную единицу
    измерения и единицу измерения, в которую нужно перевести
    через( пробел): ')
2 goo = initial.split(' ')
3 if goo[1] == 's' and goo[2] == 'm': print(str(float(goo[0])
    /60) + 'm')
4 if goo[1] == 's' and goo[2] == 'h': print(str(float(goo[0])
    /3600) + 'h')
5 if goo[1] == 'm' and goo[2] == 's': print(str(float(goo[0])
    *60) + 's')
6 if goo[1] == 'm' and goo[2] == 'h': print(str(float(goo[0])
    /60) + 'h')
7 if goo[1] == 'h' and goo[2] == 's': print(str(float(goo[0])
    *3600) + 's')
8 if goo[1] == 'h' and goo[2] == 'm': print(str(float(goo[0])
    *60) + 'm')
9 if goo[1] == 'h' and goo[2] == 'h': print(str(float(goo[0]))
    + 'h')
10 if goo[1] == 'm' and goo[2] == 'm': print(str(float(goo[0]))
    + 'm')
11 if goo[1] == 's' and goo[2] == 's': print(str(float(goo[0]))
    + 's')

```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Создается переменная `initial`, в которую пользователь вводит значение, которое нужно конвертировать;
- 2) С помощью метода `split`, переменная разделяется на три по пробелу и добавляется в список `goo`;
- 3) Перебираются все возможные комбинации букв и выводится соответствующий результат.

## 1.2 Задание 2

В данном задании необходимо написать функцию, которая рассчитывает прибыль пользователя без учета первоначально вложенной суммы. Используется сложный процент: каждый год процент начисляется на текущую сумму вклада. На рисунке 2 представлен код программы.

```

1 a = float(input('Вклад в банке в размере: '))
2 n = float(input('Срок: '))
3 deprate = 0
4 deprate3 = 3
5 deprate46 = 5
6 deprate6 = 2
7 profit = 0
8 every10deprate = 0
9
10 if a < 30000: print('Минимальный вклад - 30 000 рублей!')
11
12 elif a >= 30000:
13     every10deprate += ((a+profit)//10000) * 0.3
14     if every10deprate <= 5: deprate += ((a+profit)//10000) *
15         0.3
16     elif every10deprate >= 5: deprate += 5
17
18     if n <= 3:
19         for i in range(int(n)):
20             anew = a + profit
21             profit += anew*(deprate3+deprate)/100
22
23     elif n >= 4 and n <= 6:
24         for i in range(3):
25             anew = a + profit
26             profit += anew*(deprate3+deprate)/100
27         for i in range(int(n)-3):
28             anew = a + profit
29             profit += anew*(deprate46+deprate)/100
30
31     elif n > 6:
32         for i in range(3):
33             anew = a + profit
34             profit += anew*(deprate3+deprate)/100
35         for i in range(3):
36             anew = a + profit
37             profit += anew*(deprate46+deprate)/100
38         for i in range(int(n)-6):
39             anew = a + profit
40             profit += anew*(deprate6+deprate)/100
41 print(round(profit, 2))

```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Создается переменная a, в которую пользователь вводит размер вклада в банке;
- 2) Создается переменная n, в которую пользователь вводит срок;
- 3) Создаются переменные во всеми данными ставками;
- 4) Создается переменная profit с прибылью, которая изначально равна нулю;



- 5) С помощью условного оператора `if` проверяется меньше ли размер введенного вклада, чем 30000;
- 6) Если размер введенного вклада в банке меньше 30000, то выводится ошибка;
- 7) С помощью условного оператора `elif` проверяется больше или равен ли размер введенного вклада 30000;
- 8) Если размер введенного вклада в банке меньше 30000, то программа завершается;
- 9) Если размер введенного вклада в банке больше или равен 30000, то программа продолжается;
- 10) Вычисляются проценты за каждые 10000 тысяч рублей;
- 11) С помощью условного оператора `if`, если переменная `n` меньше или равна 3, то вычисляется соответствующая прибыль;
- 12) С помощью условного оператора `if`, если переменная `n` больше или равна 4 и меньше или равна 6, то вычисляется соответствующая прибыль;
- 13) С помощью условного оператора `if`, если переменная `n` больше 6, то вычисляется соответствующая прибыль;
- 14) Соответствующая прибыль выводится в консоль.

### 1.3 Задание 3

В данном задании необходимо написать функцию для вывода всех простых чисел в заданном диапазоне. Нужно учитывать некорректные данные (например, начало больше конца или диапазон без простых чисел) На рисунке 3 представлен код программы.

```

1 die = input('Введите начало и конец диапазона через ( пробел): ')
2 startend = die.split(' ')
3
4 def isprime(n):
5     d = 2
6     while n % d != 0:
7         d+=1
8     return d == n
9
10 if startend[0].isdigit() == False or startend[1].isdigit()
    == False: print('Ошибка! Было введено не число!')
11
12 primes = []
13
14 if startend[0].isdigit() == True and startend[1].isdigit()
    == True:
15     if int(startend[0]) > int(startend[1]): print('Ошибка!
        Начало больше конца!')
16     elif (int(startend[0]) < int(startend[1])):
17         if int(startend[0]) != 1:
18             for i in range(int(startend[0]), int(startend
                [1])+1):
19                 if isprime(i) == True:
20                     primes.append(i)
21                 i+=1
22     else:
23         for i in range(2, int(startend[1])+1):
24             if isprime(i) == True:
25                 primes.append(i)
26             i+=1
27
28 print(primes)

```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Создается переменная `die`, в которую пользователь вводит значение начала и конца диапазона;
- 2) Создается функция `isprime`, которая определяет, является ли число простым;
- 3) С помощью условного оператора `if`, если введенные значение начала и конца диапазона не являются числами, то выводится ошибка;
- 4) Создается пустой список `primes`;
- 5) С помощью условного оператора `if`, если введенные значение начала и конца диапазона являются числами, то программа продолжается;
- 6) С помощью условного оператора `if`, если введенное значение начала диапазона больше конца, то выводится ошибка;
- 7) С помощью условного оператора `elif`, если введенное значение начала диапазона меньше конца, то программа продолжается;
- 8) Отдельно высчитываются простые числа в заданном диапазоне, для диапазона, начинающегося с 1, и для остальных, и добавляются в список `primes`;
- 9) Все простые числа из заданного диапазона выводятся в консоль в виде списка `primes`.

#### 1.4 Задание 4

Необходимо реализовать функцию сложения двух матриц. На рисунке 4 представлен код решения.

```

1 import numpy as np
2 nn = int(input('Введите размер матрицы: '))
3
4 if nn < 3: print('Слишком маленький размер!')
5
6 if nn > 2:
7     m1 = []
8     m2 = []
9
10    for i in range(nn):
11        string = input('Введите строку первой матрицы: ')
12        mat = string.split(' ')
13        if len(mat) != nn:
14            print('Неверная длина!')
15            break
16        if len(mat) == nn:
17            matint = []
18            for i in mat:
19                matint.append(int(i))
20            m1.append(matint)
21
22    for i in range(nn):
23        string = input('Введите строку второй матрицы: ')
24        mat = string.split(' ')
25        if len(mat) != nn:
26            print('Неверная длина!')
27            break
28        if len(mat) == nn:
29            matint = []
30            for i in mat:
31                matint.append(int(i))
32            m2.append(matint)
33
34 if len(m1) == nn and len(m2) == nn:
35     matrix1 = np.array(m1)
36     matrix2 = np.array(m2)
37
38     print(matrix1 + matrix2)

```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Через команду `import` импортируется библиотека `numpy` как `np`;
- 2) Создается переменная `nn`, в которую пользователь вводит размер матрицы;
- 3) С помощью условного оператора `if`, если `nn` меньше 3, то выводится ошибка;
- 4) С помощью условного оператора `if`, если `nn` больше 2, то программа продолжается;
- 5) Создаются списки `m1` и `m2`;
- 6) Через цикл `for` вводятся значение первой матрицы и добавляются с список `m1`;
- 7) Через цикл `for` вводятся значение второй матрицы и добавляются с список `m2`;
- 8) С помощью условного оператора `if`, если длина `m1` равна длине `m2`, то программа продолжается;
- 9) Создается переменная `matrix1`, которой присваивается значение `m1`, в виде массива библиотеки `numpy`;
- 10) Создается переменная `matrix2`, которой присваивается значение `m2`, в виде массива библиотеки `numpy`;
- 11) Переменные `matrix1` и `matrix2` складываются и результат выводится в консоль.

## 1.5 Задание 5

В данном задании требуется написать функцию, которая определяет, является ли строка палиндромом. На рисунке 5 представлен код программы.

```

1 s = input('Введите строку: ')
2
3 ss = s.replace(' ', '').lower()
4
5 if ss == ss[::-1]: print('Да')
6 else: print('Нет')
```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Создается переменная `s`, в которую пользователь вводит строку;
- 2) Создается переменная `ss`, которой присваивается значение переменной `s`, но без пробелов и заглавных букв;
- 3) С помощью условного оператора `if`, если `ss` равна перевернутой `ss`, то в консоль выводится «Да». Иначе выводится «Нет».