

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.М. Сахарюк  
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.

- 1) На вход подается:
- 2) число (величина времени)
- 3) исходная единица измерения
- 4) единица измерения, в которую нужно перевести

Функция должна вернуть конвертированное значение

**Задание 2.** Пользователь делает вклад в банке в размере а 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)

**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:

- 1) складывать можно только матрицы одинакового размера
- 2) размер матрицы должен быть строго больше 2 (например,  $3 \times 3$ ,  $4 \times 4$  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 **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

## 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 .....	6

## 1 Выполнение работы

### 1.1 Задание 1

В данном задании требуется написать функцию для конвертирования времени. На рисунке 1 представлен код полученной программы.

```

1 TimeUnits = {'s': 1, 'm': 60, 'h': 3600, 'd': 86400}
2
3 def ConvertTime(_time, _unit):
4     for unit in TimeUnits.keys():
5         if unit in _time:
6             base_time = int(_time.replace(unit, '')) *
7             TimeUnits[unit]
8             break
9
10    return f'{base_time / TimeUnits[_unit]}{_unit}'
11
12 user_input = input('введите два значения через пробел: ')
13     split(' ')
14 print(ConvertTime(user_input[0], user_input[1]))

```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Объявляется словарь TimeUnits с соотношениями единиц времени в секундах
- 2) Определяется функция ConvertTime, принимающая время и целевую единицу
- 3) В функции из входной строки убирается единица, число переводится в секунды
- 4) Результат делится на значение целевой единицы и возвращается с её обозначением
- 5) Программа запрашивает у пользователя ввод (время и целевая единица)
- 6) Вызывает функцию и выводит результат на экран

После выполнения программы в консоль выводится сконвертированное значение времени в указанной единице.

### 1.2 Задание 2

В данном задании требуется написать функцию для расчёта прибыли по банковскому вкладу с учётом сложных процентов и зависимой ставки. На рисунке 2 представлен код полученной программы.

```

1 def CalculateVklad(_deposit, _years):
2     if int(_deposit) < 30000:
3         return 0.0
4     _years = int(_years)
5     if _years <= 3:
6         Rate = 0.03
7     elif _years <= 6:
8         Rate = 0.05
9     else:
10        Rate = 0.02
11    AddRate = (int(_deposit) // 10000) * 0.003
12    TotalRate = min(Rate + AddRate, 0.05)
13    _profit = int(_deposit) * (1 + TotalRate) ** _years -
14    int(_deposit)
15    return round(_profit, 2)
16 user_input = input('введите сумму вклада и количество лет
17 через пробел: ').split(' ')
18 print(CalculateVklad(user_input[0], user_input[1]))

```

Рисунок 2 – Листинг программы для задания 2

- 1) Проверяется, что сумма вклада не меньше 30 000 рублей
- 2) Определяется базовая ставка в зависимости от срока: 3% — до 3 лет, 5% — 4–6 лет, 2% — более 6 лет
- 3) Рассчитывается дополнительная ставка: 0.3% за каждые 10 000 рублей, но не более 5% суммарно
- 4) Итоговая ставка — минимум из (базовая + доп.) и 5%
- 5) Прибыль рассчитывается по формуле сложного процента: вклад  
(1 ставка)<sup>лет</sup> вклад
- 6) Результат округляется до двух знаков и возвращается

После выполнения программы в консоль выводится сумма заработанных процентов (прибыль) за указанный срок.

### 1.3 Задание 3

В данном задании требуется написать функцию для вывода всех простых чисел в заданном диапазоне с обработкой некорректных данных. На рисунке 3 представлен код полученной программы.

```

1 def IsPrime(_value):
2     if _value <= 1:
3         return False
4
5     limit = int(_value ** 0.5) + 1
6     for i in range(2, limit):
7         if _value % i == 0:
8             return False
9     return True
10
11 def PrintAllPrimes(_lowerLimit , _upperLimit):
12     if int(_lowerLimit) < 0 or int(_upperLimit) <= int(
13         _lowerLimit):
14         print('Error!')
15         return
16     IsPrimeFound = False
17     CompletedStr = ''
18     for digit in range(int(_lowerLimit), int(_upperLimit)):
19         if IsPrime(digit):
20             CompletedStr += str(digit) + ' '
21             IsPrimeFound = True
22     if(IsPrimeFound == False):
23         print('Error!')
24         return
25     print(CompletedStr)
26
27 user_input = input('введите два значения через пробел: ')
28 split(' ')
29 PrintAllPrimes(user_input[0], user_input[1])

```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Определяется функция IsPrime, проверяющая, является ли число простым (до корня из числа)
- 2) Определяется функция PrintAllPrimes, принимающая начало и конец диапазона
- 3) Проверяются входные данные: начало  $\geq 0$  и начало  $\leq$  конец — иначе выводится "Error!"
- 4) Перебираются все числа в диапазоне, для каждого вызывается IsPrime
- 5) Простые числа добавляются в строку через пробел
- 6) Если ни одного простого числа не найдено — выводится "Error!" иначе — строка с числами

После выполнения программы в консоль выводится строка с простыми числами в указанном диапазоне или сообщение "Error!" при некорректном вводе или отсутствии простых чисел.

#### 1.4 Задание 4

В данном задании требуется написать функцию для сложения двух квадратных матриц одинакового размера. На рисунке 4 представлен код полученной программы.

```

1 def AddMatrices(_n, _mat1, _mat2):
2     N = int(_n)
3     if N <= 2:
4         return "Error!"
5
6     Result = []
7     for i in range(N):
8         Row = []
9         for J in range(N):
10             Row.append(int(_mat1[i][J]) + int(_mat2[I][J]))
11         Result.append(Row)
12     return Result
13
14 UserInput = input('введите крч данные').strip().split()
15 N = UserInput[0]
16
17 Mat1 = []
18 Mat2 = []
19
20 for i in range(int(N)):
21     Mat1.append(input().strip().split())
22
23 for i in range(int(N)):
24     Mat2.append(input().strip().split())
25
26 Answer = AddMatrices(N, Mat1, Mat2)
27
28 if Answer == "Error!":
29     print("Error!")
30 else:
31     for Row in Answer:
32         print(" ".join(map(str, Row)))

```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Проверяется, что размер матрицы n строго больше 2 — иначе возвращается "Error!"
- 2) Создаётся пустой результирующий список Result
- 3) Внешний цикл по строкам, внутренний — по столбцам: складываются элементы с одинаковыми индексами из обеих матриц
- 4) Суммы добавляются в строки результата, строки — в общую матрицу
- 5) Функция возвращает итоговую матрицу или "Error!" при нарушении условий
- 6) Программа считывает размер и две матрицы, вызывает функцию и выводит результат построчно

После выполнения программы в консоль выводится результирующая матрица (по строкам через пробел) или сообщение "Error!" при неверном размере или формате.

## 1.5 Задание 5

В данном задании требуется написать функцию для проверки, является ли строка палиндромом (без учёта пробелов, регистра и знаков препинания). На рисунке 5 представлен код полученной программы.

```

1 import string
2
3 def IsPalindrome(_line):
4     BaseStr = str(_line).lower()
5
6     for letter in range(len(BaseStr)):
7         if BaseStr[letter] in string.punctuation:
8             BaseStr[letter] = ''
9     BaseStr = BaseStr.replace(' ', '')
10
11    if BaseStr == BaseStr[::-1]:
12        return 'Да'
13    return 'Нет'
14
15 print(IsPalindrome(input('введите строку для проверки: ')))

```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Импортируется модуль `string` для работы со знаками препинания
- 2) Стока приводится к нижнему регистру
- 3) Удаляются все пробелы и знаки препинания из строки
- 4) Сравнивается полученная строка с её обратной версией
- 5) Если совпадает — возвращается "Да иначе — "Нет"

После выполнения программы в консоль выводится "Да если строка — палиндром, или "Нет"— если нет.