

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	_____	К.И. Воробьев
Ассистент		
преподавателя	_____	М.В. Водяницкий

Задание

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

- 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 × 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:

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

Примеры: | Вход | Выход | | 'А роза упала на лапу Азора' | 'Да' | | 'Borrow or rob' | 'Да' | | 'Алфавитный порядок' | 'Нет' | ## Оформление отчета. Отчет оформляется строго по СТО. 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).

Оформление отчета

Отчет оформляется строго по СТО.

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	3
1.3	Задание 3	4
1.4	Задание 4	4
1.5	Задание 5	5

1 Выполнение работы

1.1 Задание 1

На рисунке 1 представлен код программы.

```

1 print("Задание 1")
2
3 ex1_converter_dictionary = {
4     's': 1,
5     'm': 60,
6     'h': 3600
7 }
8 def ex1_convert(convert_from, convert_to_unit):
9     return f"{float((ex1_converter_dictionary[convert_from]
10    [-1])*float(convert_from[0:len(convert_from)-1]))/
11    ex1_converter_dictionary[convert_to_unit]}}{
12    convert_to_unit}"
13 print(ex1_convert("12s", "h"),2)

```

Рисунок 1 – Код программы 1

Функция 'ex1_convert_time' принимает числовое значение, исходную единицу измерения и целевую единицу измерения. Она использует словарь 'ex1_converter_dictionary' для хранения коэффициентов перевода всех поддерживаемых единиц (секунды, минуты, часы) в секунды. Сначала значение конвертируется в секунды, а затем из секунд в целевую единицу. Результат возвращается в виде числа.

1.2 Задание 2

На рисунке 2 представлен код программы.

```

1 print("Задание 2")
2
3
4 def ex2_calculate_profit(amount, years):
5     percent = 0
6     if years <= 3:
7         percent = 3
8     elif 4 <= years <= 6:
9         percent = 5
10    else:
11        percent = 2
12
13    if amount < 30000:
14        return "Сумма вклада не может быть меньше 30000"
15    if years < 1:
16        return "Срок не может быть меньше года"
17
18    percent += min(amount//10000*0.3,5)
19
20    profit = amount
21    for _ in range(years):
22        profit += profit * percent/100
23    return round(profit-amount,2)
24 print(ex2_calculate_profit(30000,3))

```

Рисунок 2 – Код программы 2

Функция 'ex2_calculate_profit' рассчитывает прибыль по вкладу с учетом сложного процента. Она принимает начальную сумму вклада и количество лет. Сначала определяется базовая процентная ставка в зависимости от срока вклада, затем рассчитывается дополнительный бонус к ставке в зависимости от суммы вклада (с ограничением в 5

1.3 Задание 3

На рисунке 3 представлен код программы.

```

1 print("Задание 3")
2
3 def ex3_is_prime(num):
4     if num < 2:
5         return False
6     for i in range(2, int(num**0.5) + 1):
7         if num % i == 0:
8             return False
9     return True
10 def ex3_primes_in_range(a,b):
11     primes = []
12     if a>b:
13         return "Начало диапазона больше конца"
14
15     for n in range(a,b+1):
16         if ex3_is_prime(n):
17             primes.append(n)
18
19     if not primes:
20         return "В диапазоне нет простых чисел"
21     return primes
22 print(ex3_primes_in_range(1,123))

```

Рисунок 3 – Код программы 3

Функция 'ex3_primes_in_range' находит все простые числа в заданном диапазоне (включительно). Вспомогательная функция 'ex3_is_prime' проверяет, является ли число простым, путем деления на числа от 2 до квадратного корня из числа. Основная функция проверяет входные данные на корректность. Если диапазон не содержит простых чисел или входные данные некорректны, возвращается сообщение об ошибке. В противном случае возвращается массив с простыми числами.

1.4 Задание 4

На рисунке 4 представлен код программы.

```

1 print("Задание 4")
2 def ex4_matrices_add():
3     dimension = int(input())
4     m1 = []
5     for i in range(dimension):
6         m1.append([])
7         nums = input().split(" ")
8         for n in nums:
9             m1[i].append(float(n))
10
11     m2 = []
12     for i in range(dimension):
13         m2.append([])
14         nums = input().split(" ")
15         for n in nums:
16             m2[i].append(float(n))
17
18     newMatrix = []
19     for i in range(dimension):
20         newMatrix.append([])
21         for j in range(dimension):
22             newMatrix[i].append(m1[i][j]+m2[i][j])
23         print(m1[i][j]+m2[i][j],end=" ")
24     print()
25     return newMatrix
26 ex4_matrices_add()

```

Рисунок 4 – Код программы 4

Функция 'ex4_matrices_add' выполняет сложение двух квадратных матриц. Она принимает размерность 'di' и две матрицы в виде списков списков. Создается новая матрица, каждый элемент которой является суммой соответствующих элементов исходных матриц. Результат форматируется в виде строк, разделенных пробелами и переводами строки.

1.5 Задание 5

На рисунке 5 представлен код программы.

```

1 print("Задание 5")
2
3 def ex5_is_palindrome(input_string):
4     unneeded_symbols = ",!?. ;-'\"
5     formatted_string = input_string.lower()
6     for s in unneeded_symbols:
7         formatted_string = formatted_string.replace(s,"")
8     formatted_string = formatted_string.strip()
9     return formatted_string == formatted_string[::-1]
10
11 input_str = "abcdeftregfer"
12 print(ex5_is_palindrome(input_str))

```

Рисунок 5 – Код программы 5

Функция 'ex5_is_palindrome' определяет, является ли входная строка палиндромом. Она игнорирует регистр, пробелы и знаки препинания. Сначала строка переводится в нижний регистр, затем удаляются все лишние символы. После этого очищенная строка

сравнивается со своей перевернутой версией. Если они совпадают, возвращается True, иначе - False.