

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 №5  
по дисциплине  
«Информатика и программирование»

Студент  
гр. БИН-25-3 \_\_\_\_\_ А. Е. Филатов  
Ассистент  
преподавателя \_\_\_\_\_ М.В. Водяницкий

## Задание

Выполнить задания и оформить отчет по стандартам ВВГУ.

**Задание 1.** «<<<<<<< Temporary merge branch 1 Дан список из 10 различных целых чисел. Необходимо найти в нем число 3 и заменить на 30.

**Задание 2.** Дан список из 5 целых чисел. Необходимо превратить его в список квадратов этих чисел.

**Задание 3.** Имеется список различных целых чисел. Программа должна найти наибольшее из чисел списка и разделить его на длину списка. ===== Дан список из 10 различных целых чисел. Необходимо найти в нем число 3 и заменить на 30.

**Задание 2.** Дан список из 5 целых чисел. Необходимо превратить его в список квадратов этих чисел.

**Задание 3.** Имеется список различных целых чисел. Программа должна найти наибольшее из чисел списка и разделить его на длину списка. >>>>>>> Temporary merge branch 2

**Задание 4.** Имеется кортеж из нескольких произвольных элементов. Необходимо этот кортеж отсортировать. Если хотя бы один элемент не является числом, то кортеж остается неизменным.

**Задание 5.** Имеется словарь товаров в магазине. Необходимо найти товар с минимальной и максимальной ценой.

**Задание 6.** Имеется список произвольных элементов. Необходимо на основе этого списка создать словарь, где каждый элемент списка будет и ключом, и значением.

**Задание 7.** Имеется словарь перевода английских слов на русский, где ключ английского слова, значение - русского. Необходимо реализовать программу которая получает на ввод русское слово и результатом выдает перевод на английский.

**Задание 8.** Реализовать игру Камень-Ножницы-Бумага-Ящерица-Спок. Программа должна запрашивать у пользователя ввод одного из вариантов. Второй вариант случайно генерирует сама программа и возвращает победителя.

Правила игры следующие:

- Ножницы режут бумагу
- Бумага покрывает камень
- Камень давит ящерицу
- Ящерица отравляет Спока
- Спок ломает ножницы

- Ножницы обезглавливают ящерицу
- Ящерица съедает бумагу
- Бумага подставляет Спока
- Спок испаряет камень
- Камень разбивает ножницы

**Задание 9.** Дан список слов - например:

- ‘[”яблоко ”груша ”банан ”киви ”апельсин ”ананас”]’

Необходимо создать новый словарь, где:

- Ключом будет первая буква слова
- Значением - список всех слов, начинающихся с этой буквы

Пример результата:

’я’: [’яблоко’], ’г’: [’груша’], ’б’: [’банан’], ’к’: [’киви’], ’а’: [’апельсин’, ’ананас’]

**Задание 10.** Дан список кортежей, где каждый кортеж содержит имя студента и его оценки, например:

[(”Анна [5, 4, 5]), (”Иван [3, 4, 4]), (”Мария [5, 5, 5])]

Необходимо:

- Создать словарь, где ключ - имя студента, значение - средняя оценка
- Найти студента с наивысшей средней оценкой

## Содержание

## 1 Выполнение работы

### 1.1 Задание 1

Сначала создадим список `lists`, заменяем число 3 на 30, а после выводим результат.

На рисунке 1 представлен код программы.

```
1 lists=[1,2,142,4,5,3,7,8,9,10]
2 lists[lists.index(3)] = 30
3 print(lists)
```

Рисунок 1 – Листинг программы для задания 1

- 1) создадим список из 10 различных целых чисел;
- 2) ищем в списке число 3 и заменяем его на 30;
- 3) выводим результат

### 1.2 Задание 2

Создаем список из 5 различных чисел, а дальше нам нужно превратить его в список квадратов этих чисел. На рисунке 2 представлен код программы.

```
1 lis=[1,2,3,4,5]
2 print([x**2 for x in lis])
```

Рисунок 2 – Листинг программы для задания 2

- 1) создаем список из 5 различных чисел;
- 2) используем генератор списков для возведения каждого числа в квадрат и выводим результат.

### 1.3 Задание 3

Нам требуется создать произвольный список чисел, а далее программа должна найти наибольшее из чисел списка и разделить его на длину списка. После выводим в консоль результат. На рисунке 3 представлен код программы.

```
1 lis=[1,2,3,4,6,6,19,8,9,0,17,5]
2 print(max(lis)/len(lis))
```

Рисунок 3 – Листинг программы для задания 3

- 1) создаем произвольный список чисел;
- 2) находим максимальное число из списка, далее делим на длину списка и выводим результат пользователю.

### 1.4 Задание 4

Создаем функцию которая проверяет состоит ли кортеж только из чисел, если да, то сортируем числа от меньшего к большему, если нет, то кортеж остается неизменным. На рисунке 4 представлен код решения.

```

1 def sort(x):
2     if all(isinstance(item, (int, float)) for item in x):
3         return tuple(sorted(x))
4     return x
5 lis1=[3,2,5,6,1,5]
6 lis2=['a','f',3,5,'h',4]
7 print(sort(lis1))
8 print(sort(lis2))

```

Рисунок 4 – Листинг программы для задания 4

- 1) объявляем функцию sort с параметром x;
- 2) проверка, все ли элементы в x - числа (int или float);
- 3) если возвращает значение True, то сортируем кортеж и возвращаем отсортированный кортеж;
- 4) если возвращает значение False, то возвращаем кортеж без изменений;
- 5) проверка работы функции значением True;
- 6) проверка работы функции с значением False.

### 1.5 Задание 5

Создаем функцию которая получает на ввод словарь и выведет пользователю товар с максимальной и минимальной ценой. На рисунке 5 представлен код программы.

```

1 def help(x):
2     min1=min(products, key=products.get)
3     max1=max(products, key=products.get)
4     return min1, max1
5 products={"apple":100,"banana":80,"orange":120,"grape":90}
6 print(help(products))

```

Рисунок 5 – Листинг программы для задания 5

- 1) объявляем функцию help с параметром x;
- 2) ищем товара с минимальной ценой с помощью функции min с использованием функции для получения значения по ключу и ложим значение в переменную min1;
- 3) находит ключ с максимальным значением в словаре products и ложим в переменную max1;
- 4) возвращаем переменные min1 и max1;
- 5) создаем словарь товаров products;
- 6) вызываем функцию help с аргументом products и ложим возвращаемые значения.

### 1.6 Задание 6

Имеется список произвольных элементов, программа на основе данного списка создает словарь, где каждый элемент списка будет ключом и значением. На рисунке 6 представлен код программы.

```

1 list1=[1, "hello", 3.14, True, (1, 2)]
2 def help(x):
3     dict1={}
4     for i in x:
5         dict1[i]=i
6     return dict1
7 print(help(list1))

```

Рисунок 6 – Листинг программы для задания 6

- 1) создаем список list1;
- 2) объявляем функцию help с параметром x;
- 3) создаем пустой словарь dict1;
- 4) проходимся списком по всем элементам списка x;
- 5) добавление в словарь пары ключ-значение, где и ключ и значение - сам элемент;
- 6) возврат полученного словаря;
- 7) вызов функции help с аргументом list1 и вывод результата.

## 1.7 Задание 7

Сперва запрашиваем три числа у пользователя. Далее ищем минимального значения через условные операторы. Первая проверка: Проверяет, является ли a меньше или равным b И a меньше или равным c. Если оба условия истинны, a - наименьшее число. Вторая проверка: Выполняется, если первое условие ложно. Проверяет, является ли b меньше или равным a И b меньше или равным c. Если истинно, b - наименьшее число. Третий случай: else: Срабатывает, если оба предыдущих условия ложны. Значит, c - наименьшее число. Найденное наименьшее число выводится с использованием f-строки. На рисунке 7 представлен код программы.

```

1 def eng_rus(x,word):
2     for eng, rus in x.items():
3         if rus==word:
4             return eng
5     return None
6 d = {"apple": "яблоко", "banana": "банан", "orange": "апельсин",
7     "cat": "кот", "dog": "собака", "house": "дом", "car": "
    машина"}
8 print(eng_rus(d, 'яблоко'))

```

Рисунок 7 – Листинг программы для задания 7

## 1.8 Задание 8

Получаем исходную сумму покупки от пользователя для дальнейших расчетов. - преобразует введенную строку в число с плавающей точкой. Размер скидки зависит от суммы покупки: сумма до 1000р. скидка 0. На рисунке 8 представлен код программы.

```

1 import random
2 x=input("Введите ваш выбор ножницы(, бумага, камень, ящерица,
   спок): ")
3 def game(x):
4     rules = {'ножницы': ['бумага', 'ящерица'], 'бумага': ['
   камень', 'спок'], 'камень': ['ножницы', 'ящерица'], 'ящерица':
   ['бумага', 'спок'], 'спок': ['ножницы', 'камень']}
5     y=random.choice(list(rules.keys()))
6     print("Компьютер выбрал:", y)
7     if x == y:
8         return "Ничья!"
9     elif y in rules[x]:
10        return "Вы выиграли."
11    else:
12        return "Компьютер выиграл!"
13 print(game(x))

```

Рисунок 8 – Листинг программы для задания 8

### 1.9 Задание 9

Создаем функцию time с одним параметром num, параметр num представляет час суток (0-23). После проверяем валидность входных данных:  $0 > \text{num} < 23$ ; часы не могут быть отрицательными/ в сутках не более 23 часов. При невалидном вводе выводится сообщение об ошибке. Если данные корректны, то проверяется принадлежность к времени суток с помощью оператора in и заранее определенных списков с часами каждого из времени суто. На рисунке 9 представлен код программы.

```

1 words = ["яблоко", "груша", "банан", "киви", "апельсин", "
   ананас"]
2 def dict(x):
3     first = set(word[0] for word in x)
4     return {letter: [word for word in x if word[0] == letter
   ]
5             for letter in first}
6 result_dict = dict(words)
7 print(dict(words))

```

Рисунок 9 – Листинг программы для задания 9

### 1.10 Задание 10

Получаем число от пользователя с помощью input(), int() преобразовывает строку в целое число. При вводе нечисловых данных выводим ошибку. После того, как мы убедились в корректности данных, начинаем обрабатывать число, а именно: если число  $< 1$  или четное, то оно уже не простое, кроме 2. Далее запускаем цикл, в котором мы будем искать делители нашего числа, если найдутся еще кроме 1 и самого числа,



то выводим в консоль, что число составное, в ином случае - простое. На рисунке 10 представлен код программы.

```
1 students = [("Анна", [5, 4, 5]), ("Иван", [3, 4, 4]), ("
    Мария", [5, 5, 5])]
2 average_grades = {}
3 for name, grades in students:
4     average = sum(grades) / len(grades)
5     average_grades[name] = average
6 best_student = max(average_grades, key=average_grades.get)
7 best_average = average_grades[best_student]
8 print(f"{best_student} имеет наивысший средний балл: {
    best_average}")
```

Рисунок 10 – Листинг программы для задания 10

Спасибо за внимание !