

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	_____	Л.А. Сидоров
Ассистент		
преподавателя	_____	М.В. Водяницкий

Задание

Выполнить задания и оформить отчет по стандартам ВВГУ.

Задание 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.

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 - его средняя оценка
- Найти студента с наибольшей средней оценкой и вывести его имя и средний

балл

Пример результата:

Мария имеет наивысший средний балл: 5.0

Содержание

1	Выполнение работы.....	3
1.1	Задание 1	3
1.2	Задание 2	3
1.3	Задание 3	3
1.4	Задание 4	4
1.5	Задание 5	4
1.6	Задание 6	6
1.7	Задание 7	6
1.8	Задание 8	6
1.9	Задание 9	7
1.10	Задание 10	7

1 Выполнение работы

1.1 Задание 1

Процедура на вход принимает указатель на первый элемент массива и его размер. Далее в цикле `for` мы пробегаемся по элементам массива в поисках числа 3, если находим, то меняем значение на 30, в ином случае массив остается прежним. На рисунке 1 представлен код полученной программы.

```
1 void task1(int *arrP, int size)
2 {
3     for (int i = 0; i < size; i++)
4     {
5         if (*(arrP+i) == 3) *(arrP+i) = 30;
6     }
7 }
```

Рисунок 1 – Листинг программы для задания 1

1.2 Задание 2

Процедура на вход принимает указатель на первый элемент массива и его размер. Далее мы пробегаемся с помощью цикла `for` по массиву и возводим все его элементы в квадрат.

```
1 void task2(int *arrP, int size)
2 {
3     for (int i = 0; i < size; i++)
4     {
5         *(arrP+i) *= *(arrP+i);
6     }
7 }
```

Рисунок 2 – Листинг программы для задания 2

1.3 Задание 3

Функция на вход принимает указатель на первый элемент массива и его размер. Далее заводим переменные `maxNum` типа `integer`, инициализируем нулевым элементом массива. Именно ее мы в цикле `for`, пробая по массиву, будем сравнивать с его элементами и искать наибольшее значение. После нахождения функция возвращает значение, деленное на размер массива. На рисунке 3 представлен код программы.

```

1 float task3(int *arrP, int size)
2 {
3     int maxNum = *arrP;
4     int num;
5
6     for (int i = 0; i < size; i++)
7     {
8         if (*(arrP+i) > maxNum)
9         {
10            maxNum = *(arrP+i);
11        }
12    }
13
14    return (maxNum/size);
15 }

```

Рисунок 3 – Листинг программы для задания 3

1.4 Задание 4

Функция на вход принимает ссылку на кортеж. Далее в переменную size мы записываем его размер. После в цикле for пробегаемся по кортежу, проверяя его элементы на тип данных, в случае, если находим не число, то возвращаем кортеж без изменений. В ином случае извлекаем его элементы в список и сортируем. В конце конвертируем список в кортеж и возвращаем измененную версию. На рисунке 4 представлен код решения.

```

1 def task4(tupl):
2     size = len(tupl)
3
4     for i in range (0, size):
5         if (type(tupl[i]) != int):
6             return tupl
7
8     listOfNums = list(tupl)
9     sortedListOfNums = sorted(listOfNums)
10
11    return tuple(sortedListOfNums)

```

Рисунок 4 – Листинг программы для задания 4

1.5 Задание 5

Функция принимает на вход словарь с продуктами и ценами. Далее из этого словаря мы извлекаем в отдельные списки продукты и цены по соответствующим индексам. После находим в списке цен индексы с минимальной и максимальной ценами. В конце возвращаем словарь, подобный словарю, однако состоящий лишь из продуктов с минимальной и максимальной ценами. На рисунке 5 представлен код программы.

```

1 def getIndexMin(ls):
2     j = 0;
3     minNum = ls[j]
4     size = len(ls)
5     for i in range (0, size):
6         if (ls[i] < minNum):
7             minNum = ls[i]
8             j = i
9     return j
10
11 def getIndexMax(ls):
12     j = 0;
13     maxNum = ls[j]
14     size = len(ls)
15     for i in range (0, size):
16         if (ls[i] > maxNum):
17             maxNum = ls[i]
18             j = i
19     return j
20
21 def task5(dictProducts):
22     lsProducts = list(dictProducts.keys())
23     lsCostProducts = list(dictProducts.values())
24
25     iProductMinCost = getIndexMin(lsCostProducts)
26     iProductMaxCost = getIndexMax(lsCostProducts)
27
28     return {
29         lsProducts[iProductMinCost] : lsCostProducts[
30             iProductMinCost],
31         lsProducts[iProductMaxCost] : lsCostProducts[
32             iProductMaxCost]
33     }

```

Рисунок 5 – Листинг программы для задания 5

1.6 Задание 6

Функция получает на вход список. В переменную size мы запишем его размер и создадим пока пустой словарь. Далее в цикле for пробежимся по списку и каждому элементу списка создадим в словаре объект с ключем и значением, соответствующими содержанию элемента списка. В конце она возвращает полученный словарь. На рисунке 6 представлен код программы.

```

1 def task6(ls):
2     size = len(ls)
3     dictionary = {}
4
5     for i in range(0, size):
6         dictionary[ls[i]] = ls[i]
7
8     return dictionary

```

Рисунок 6 – Листинг программы для задания 6

1.7 Задание 7

Функция на вход принимает словарь из английских слов и их русским переводом. Далее из этого словаря мы извлекаем в отдельные списки английские и русские слова по

соответствующим индексам. После пробегаемся по русскому списку в поисках указанного слова. Если находим, то возвращаем элемент из английского списка с нужным индексом. В ином случае возвращаем None. На рисунке 7 представлен код программы.

```

1 def task7(dictEnRu, RuWord):
2     lsEnWords = list(dictEnRu.keys())
3     lsRusWords = list(dictEnRu.values())
4
5     sizeLsRusWords = len(lsRusWords)
6
7     for i in range(0, sizeLsRusWords):
8         if (lsRusWords[i].lower() == RuWord.lower()):
9             return lsEnWords[i]
10
11     return None

```

Рисунок 7 – Листинг программы для задания 7

1.8 Задание 8

Функция на вход принимает ход игрока. Далее мы создаем список из возможных ходов и словарь с правилами, где указан ход в качестве ключа и то, при каких обстоятельствах он выигрышный в качестве значения. После генерируется ход программы. В конце мы проверяем, если ходы одинаковые, то возвращаем ничью, если ход программы содержится в значениях словаря под ходом игрока, то возвращаем, что игрок победил, в ином случае, возвращаем, что он проиграл. На рисунке 8 представлен код программы.

```

1 def task8(playerMove):
2     playerMove = playerMove.lower()
3     moves = ("камень", "ножницы", "бумага", "ящерица", "спок")
4     programMove = random.randint(0, 4);
5     dictPlayRules = {
6         "камень": ["ножницы", "ящерица"],
7         "ножницы": ["бумага", "ящерица"],
8         "бумага": ["камень", "спок"],
9         "ящерица": ["бумага", "спок"],
10        "спок": ["камень", "ножницы"]
11    }
12
13    if (playerMove in moves):
14        if (programMove == playerMove):
15            return "Ничья"
16        elif (moves[programMove] in dictPlayRules[playerMove]):
17            return "Вы выиграли !"
18        else:
19            return "Вы проиграли !"
20    else:
21        return "Command not found"

```

Рисунок 8 – Листинг программы для задания 8

1.9 Задание 9

Функция на вход принимает список слов. Далее мы создаем пустой словарь, в который будем записывать слово, в качестве значения и первую букву как ключ. После циклом for пробегаемся по списку и смотрим на каждый элемент, проверяем, есть ли он

уже в словаре, если нет, то добавляем ключ и значение. В конце возвращаем полученный словарь. На рисунке 9 представлен код программы.

```

1 def task9(ls):
2     size = len(ls)
3     dictionary = {}
4
5     for i in range(0, size):
6         word = ls[i].lower()
7         firstLitWord = ls[i][0].lower()
8
9         if (lit in dictionary):
10             if (word not in dictionary[lit]):
11                 dictionary[lit].append(word)
12         else:
13             dictionary[lit] = [word]
14
15     return dictionary

```

Рисунок 9 – Листинг программы для задания 9

1.10 Задание 10

Функция на вход принимает список картежей с именами и оценками учеников. В цикле for мы пробегаемся по картежам списка, получаем имя и оценки ученика. Далее записываем имя в качестве ключа и средний бал в качестве значения в созданный словарь. По пути программа ищет наивысший балл. В результате функция возвращает строку с именем и высшим балом ученика. На рисунке 10 представлен код программы.

```

1 def getMeanValue(lsNums):
2     size = len(lsNums)
3     res = 0
4     for i in range(0, size):
5         res += lsNums[i]
6
7     return res/size
8
9 def task10(ls):
10    size = len(ls)
11    dictionary = {}
12    maxMeanMark = 0
13    j = 0
14
15    for i in range(0, size):
16        name = ls[i][0]
17        marks = ls[i][1]
18
19        meanMark = getMeanValue(marks)
20        dictionary[name] = meanMark
21
22        if (meanMark > maxMeanMark):
23            maxMeanMark = meanMark
24            j = i
25
26    return f"{ls[j][0]} имеет наивысший балл: {maxMeanMark:.2f}"

```

Рисунок 10 – Листинг программы для задания 10

Примечание: 1 - 3 решения представляют из себя отдельные процедуры, обрабатывающие и именуяющие полученные данные. 4 - 10 решения представляют из себя отдельные функции, возвращающие то, что просят.