**Федеральное государственное образовательное**

**бюджетное учреждение высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**ИНСТИТУТ ЗАОЧНОГО И ОТКРЫТОГО ОБРАЗОВАНИЯ**

**ДЕПАРТАМЕНТ АНАЛИЗА ДАННЫХ,**

**ПРИНЯТИЯ РЕШЕНИЙ И ФИНАНСОВЫХ ТЕХНОЛОГИЙ**

**ПРАКТИЧЕСКАЯ РАБОТА № 5**

**по дисциплине «Алгоритмы и структуры данных в языке Python»**

**Тема: списки и обработка исключений**

Направление подготовки - 09.03.03 «Прикладная информатика»

Студент(ка) **Белоусов А.А.**

(Ф.И.О.)

Курс1 № группы **ЗБ-ПИ1-1**

Преподаватель **Петросов Д.А.**

(Ф.И.О.)

Москва 2020

Задания

**Задание 1**

Нaпишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его cоседей. Для элeментов списка, являющиxся крайними, одним из соседей считается элемент, находящий на противоположном конце этого списка. Например, если на вход подаётся cписок «1 3 5 6 10», то на выход ожидается cписок «13 6 9 15 7». Если на вход пришло только однo число, надо вывести его же. Вывoд должен содержать одну строку с чиcлами новoго списка, разделёнными пробeлом.

**Задание 2**

Нaпишите прогрaмму, котoрая принимает на вход спиcок чисел в одной cтроке и выводит на экран в oдну строкy значения, котoрые повторяются в нём бoлее одного раза. Выводимые числа не дoлжны повторяться, пoрядок их вывода может быть произвольным. Нaпример: 4 8 0 3 4 2 0 3.

**Задание 3**

Выполните oбработку элементов прямоoгольной матрицы A, имеющей N строк и M столбцов. Все элeменты имeют целый тип. Дано целое число H. Опрeделите, какие столбцы имeют хотя бы однo такое число, а какие не имeют.

**Задание 4**

Список задается пользователем с клавиатуры. Определите, является ли список симметричным.

**Задание 5**

Список задается пользователем с клавиатуры. Определите, можно ли удалить из списка каких-нибудь два элемента так, чтобы новый список оказался упорядоченным.

**Задание 6**

Список задается пользователем с клавиатуры. Определите, сколько различных значений содержится в списке.

**Задание 7**

Список задается пользователем с клавиатуры. Удаление из списка элементов, значения которых уже встречались в предыдущих элементах.

**Задание 8**

Пользователь вводит упорядоченный список книг (заданной длины по алфавиту). Добавить новую книгу, сохранивупорядоченность списка по алфавиту

**Задание 9**

Дан список целых чисел. Упорядочьте по возрастанию только:

а) положительные числа;

б) элементы с четными порядковыми номерами в списке.

**Задание 10**

Даны два списка. Определите, совпадают ли множества их элементов.

**Обработка следующих исключений:**

1. Exception
2. AttributeError
3. IndexError
4. KeyError
5. ValueError

Код программы

*# Задание №1***def** sumNeighbors(strlist):  
 numbers = [int(num) **for** num **in** strlist.split(**' '**)]  
 **if** len(numbers) == 1:  
 **return** str(numbers[0])  
 **return ' '**.join([str(numbers[i - 1] + numbers[(i + 1) % len(numbers)]) **for** i **in** range(len(numbers))])  
  
  
*# Задание №2***def** repeatElements(strlist):  
 numbers = [int(num) **for** num **in** strlist.split(**' '**)]  
 **return ' '**.join(set([str(i) **for** i **in** numbers **if** numbers.count(i) > 1]))  
  
  
*# Задание №3***def** colsWithNumber(matrix, number):  
 cols = []  
 **for** j **in** range(len(matrix[0])):  
 **for** i **in** range(len(matrix)):  
 **if** matrix[i][j] == number:  
 cols.append(j)  
 **break  
 return** cols  
  
  
*# Задание №4***def** listSimmetric(strlist):  
 numbers = [int(num) **for** num **in** strlist.split(**' '**)]  
 **return** all([numbers[i] == numbers[-(i + 1)] **for** i **in** range(int(len(numbers) / 2))])  
  
  
*# Задание №5***def** listCheckOrder(data, countdel=2):  
 **if** len(data) < countdel + 2:  
 **return True** count = 0  
 j = 0  
 datalist = data[:]  
 **while** j < len(datalist) - 1:  
 **for** i **in** range(len(datalist) - 1):  
 **if** datalist[i] > datalist[i + 1]:  
 *# print(datalist[i+1])* count += 1  
 **del** datalist[i + 1]  
 *# print(datalist)* **break** j += 1  
  
 **if** count <= countdel:  
 **return True** count = 0  
 j = 0  
 datalist = data[:]  
 **while** j < len(datalist) - 1:  
 **for** i **in** range(len(datalist) - 1):  
 **if** datalist[i] < datalist[i + 1]:  
 count += 1  
 **del** datalist[i + 1]  
 **break** j += 1  
 **return** count <= countdel  
  
  
*# Задание №6***def** countUniqueElements(strlist):  
 **return** len(set([num **for** num **in** strlist.split(**' '**)]))  
  
  
*# Задание №7***def** uniqueElements(strlist):  
 numbers = [num **for** num **in** strlist.split(**' '**)]  
 i = 0  
 **while** (i < len(numbers) - 1):  
 j = i + 1  
 **while** (j < len(numbers)):  
 **if** numbers[i] == numbers[j]:  
 **del** numbers[j]  
 **else**:  
 j += 1  
 i += 1  
  
 **return ' '**.join(numbers)  
  
  
**def** equalSetLists(list1, list2):  
 **return** set(list1) == set(list2)  
  
  
**def** function2():  
 **while** (**True**):  
 N = int(input(**'Введите N: '**))  
 **try**:  
 **if** N < 1:  
 **raise** Exception(**"Число не являеться натуральным"**)  
 **except** Exception **as** e:  
 print(e)  
 print(**"Повторите ввод"**)  
 **continue  
 break** M = S = 1  
 **while** (S <= N):  
 M += 1  
 S += M \*\* 2  
 **return** M - 1  
  
  
**def** getCPUbyModel(CPUs, model):  
 **for** (key, value) **in** CPUs.items():  
 **if** key[1] == model:  
 **return** key, value  
 **return None  
  
  
def** function3():  
 a = float(input(**'Введите a: '**))  
 **if** a <= 0:  
 **raise** ValueError(**"a должна быть положительным числом"**)  
 c = float(input(**'Введите c: '**))  
 **if** c <= 0:  
 **raise** ValueError(**"c должна быть положительным числом"**)  
 R = float(input(**'Введите R: '**))  
 **if** R <= 0:  
 **raise** ValueError(**"Радиус должен быть положительным числом"**)  
 x = float(input(**'Введите x: '**))  
 **if** x < -R:  
 **return** 0  
 **elif** x < 0:  
 **return** math.sqrt(R \*\* 2 - x \*\* 2)  
 **else**:  
 **return** x \* a / c  
  
  
**def** main():  
 print(**"Задание №1"**)  
 print(sumNeighbors(**"7"**))  
 print(sumNeighbors(**"7 8"**))  
 print(sumNeighbors(**"1 3 5 6 10"**))  
 print()  
  
 print(**"Задание №2"**)  
 print(repeatElements(**"7"**))  
 print(repeatElements(**"-4 8"**))  
 print(repeatElements(**"-4 -4 8 7 9 2 -7 7 -11 -4 7 -7"**))  
 print(repeatElements(**"4 8 0 3 4 2 0 3"**))  
 print()  
  
 print(**"Задание №3"**)  
 print(colsWithNumber([[0, 0, -1, 9, 0],  
 [0, 7, 0, 12, -4],  
 [17, 10, 0, 17, 10],  
 [12, -5, 0, 0, 7],  
 [7, -11, 0, 77, 0]], 7))  
 print(colsWithNumber([[]], -7))  
 print(colsWithNumber([[0, 0, -1, 9, 0, 12],  
 [0, 7, 0, 12, -4, -4],  
 [17, 10, 0, 17, 10, 0],  
 [12, -5, 0, 0, 7, 9],  
 [7, -11, 0, 77, 0, 11]], -4))  
 print(colsWithNumber([[0, 0, -1, 9, 0],  
 [0, 7, 0, 12, -4],  
 [17, 10, 0, 17, 10]], 17))  
 print(colsWithNumber([[0, 0, -1, 9, 0],  
 [0, 7, 0, 12, -4],  
 [17, 10, 0, 17, 10]], 8))  
 print()  
  
 print(**"Задание №4"**)  
 print(listSimmetric(input(**"Введите последовательность чисел через пробел: "**)))  
 print(listSimmetric(**"1 -2 3 -2 7"**))  
 print(listSimmetric(**"1"**))  
 print(listSimmetric(**"1 -2 3 -2"**))  
 print(listSimmetric(**"1 -2 -2 1"**))  
 print()  
  
 print(**"Задание №5"**)  
 print(listCheckOrder(list(map(**lambda** x: int(x), input(**"Введите последовательность чисел через пробел: "**).split()))))  
 print(listCheckOrder([1, 2, 3]))  
 print(listCheckOrder([-1, -2, -3]))  
 print(listCheckOrder([-1, -2, -3, -3]))  
 print(listCheckOrder([-1, -2, -3, -4, 1, 2, 3]))  
 print(listCheckOrder([1, 2, 3, 4, -1, -2, -3]))  
 print(listCheckOrder([2, 3, 4, 1, 3, 5]))  
 print(listCheckOrder([-2, -3, -4, -1, -3, -5]))  
 print()  
  
 print(**"Задание №6"**)  
 print(countUniqueElements(input(**"Введите последовательность чисел через пробел: "**)))  
 print(countUniqueElements(**"1 -2 3"**))  
 print(countUniqueElements(**"1 -2 7 3 -2 1 7 7"**))  
 print()  
  
 print(**"Задание №7"**)  
 print(uniqueElements(input(**"Введите последовательность чисел через пробел: "**)))  
 print(uniqueElements(**"1 -2 3"**))  
 print(uniqueElements(**"1 -2 7 3 -2 1 7 7"**))  
 print()  
  
 print(**"Задание №8"**)  
 books = list(  
 map(**lambda** s: s.strip(), input(**"Введите упорядоченный по алфавиту список книг через запятую:\n"**).split(**','**)))  
 book = input(**"Введите ещё одну книгу: "**)  
 books.append(book)  
 books.sort()  
 print(**"Упорядоченный по алфавиту список книг: "**)  
 **for** book **in** books:  
 print(book)  
 print()  
  
 print(**"Задание №9"**)  
 numbers = [-1, 0, 7, 18, 15, -4, 7, 11, 0, -100, 100, 20, 17, -18, 17, 16, 24, 27]numbers1 = numbers[:]  
 print(**"Исходный список: "**)  
 print(numbers)  
 **for** i **in** range(0, len(numbers) - 1):  
 **for** j **in** range(i + 1, len(numbers)):  
 **if** numbers[i] > 0 **and** numbers[j] > 0 **and** numbers[i] > numbers[j]:  
 numbers[i], numbers[j] = numbers[j], numbers[i]  
 print(**"Сортировка положительных значений: "**)  
 print(numbers)  
 numbers = numbers1  
 **for** i **in** range(0, len(numbers) - 1, 2):  
 **for** j **in** range(i + 2, len(numbers), 2):  
 **if** numbers[i] > numbers[j]:  
 numbers[i], numbers[j] = numbers[j], numbers[i]  
 print(**"Сортировка чётных элементов: "**)  
 print(numbers)  
 print()  
  
 print(**"Задание №10"**)  
 print(equalSetLists(numbers, numbers[:]))  
 print(equalSetLists([1, 2, 3, 4, -4, 4], [1, 2, 3, 4, -4, 4, 4, 1]))  
 print(equalSetLists([1, 2, 3, 4, -4, 4], [1, 2, 3, 4, -4, 4, 4, -1]))  
 print()  
  
 print(**"Обработка исключений"**)  
 print(**"Exception"**)  
 print(function2())  
 print()  
  
 print(**"AttributeError"**)  
 **try**:  
 x = ++function2.result  
 **except** AttributeError:  
 print(**"У функции нет свойства result"**)  
 print()  
  
 print(**"IndexError"**)  
 **try**:  
 print(getCPUbyModel({**"a"**: 1}, **"Ryzen 5 1700"**))  
 **except** IndexError:  
 print(**"Неправильный формат данных"**)  
 print()  
  
 print(**"KeyError"**)  
 processors = {(**"Ice Lake"**, **"1068G7"**): (4, 8), (**"Ice Lake"**, **"1005G1"**): (2, 4), (**"Zen"**, **"Ryzen 5 1600X"**): (6, 12),  
 (**"Zen"**, **"Ryzen 7 1700"**): (8, 16), (**"Kaby Lake"**, **"7640X"**): (4, 4)}  
 **try**:  
 print(processors[**"Ryzen 5 1700"**])  
 **except** KeyError:  
 print(**"Неправильный поиск по словарю"**)  
 print()  
  
 print(**"ValueError"**)  
 **while** (**True**):  
 **try**:  
 print(function3())  
 **except** ValueError **as** e:  
 print(e)  
 print(**"Повторите ввод параметров"**)  
 **continue  
 break** print()  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()

Результат выполнения программы

**Задание №1**

7

16 14

13 6 9 15 7

**Задание №2**

-4 7 -7

4 3 0

**Задание №3**

[0, 1, 4]

[]

[4, 5]

[0, 3]

[]

**Задание №4**

Введите последовательность чисел через пробел: 1 -2 3 -2 1

True

False

True

False

True

**Задание №5**

Введите последовательность чисел через пробел: -7 -8 -9 0 -10 1 -11

True

True

True

True

False

False

True

True

**Задание №6**

Введите последовательность чисел через пробел: 1 -2 -2 3 4 -4 -2 1 7

6

3

4

**Задание №7**

Введите последовательность чисел через пробел: 1 -2 3 -2 4 3 7 -4 7

1 -2 3 4 7 -4

1 -2 3

1 -2 7 3

**Задание №8**

Введите упорядоченный по алфавиту список книг через запятую:

Игра в имитацию, Тень Бонапарта, Практикум по высшей математике

Введите ещё одну книгу: Ревизор

Упорядоченный по алфавиту список книг:

Игра в имитацию

Практикум по высшей математике

Ревизор

Тень Бонапарта

**Задание №9**

Исходный список:

[-1, 0, 7, 18, 15, -4, 7, 11, 0, -100, 100, 20, 17, -18, 17, 16, 24, 27]

Сортировка положительных значений:

[-1, 0, 7, 7, 11, -4, 15, 16, 0, -100, 17, 17, 18, -18, 20, 24, 27, 100]

Сортировка чётных элементов:

[-1, 0, 0, 18, 7, -4, 7, 11, 15, -100, 17, 20, 17, -18, 24, 16, 100, 27]

**Задание №10**

True

True

False

**Обработка исключений**

**Exception**

Введите N: 0

Число не являеться натуральным

Повторите ввод

Введите N: 14

3

**AttributeError**

У функции нет свойства result

**IndexError**

Неправильный формат данных

**KeyError**

Неправильный поиск по словарю

**ValueError**

Введите a: -1

a должна быть положительным числом

Повторите ввод параметров

Введите a: 0

a должна быть положительным числом

Повторите ввод параметров

Введите a: 1

Введите c: 0

c должна быть положительным числом

Повторите ввод параметров

Введите a: 1

Введите c: 2

Введите R: 3

Введите x: 7

3.5

Вывод

В ходе выполнения практической работы были получены навыки работы со списками в языке Python, а также опыт обработки исключений.