Лабораторная работа 5

Напишите программу, используяминимум 5функциидля работы со списком.В виде списка пусть каждый студент предложитсвое резюме. И будет работать с этим списком.

Решите следующие задачи:

1. Напишите программу, в которой предлагается вводить учащихся различных групп, посещающих секции по разным предметам. Требуется упорядочить список по различным категориям. Вывести результат на экран.

# Создаем пустой словарь для хранения студентов по группам  
students = {}  
  
# Запускаем цикл для ввода данных о студентах  
while True:  
 group = input("Введите группу студента (или 'конец' для завершения ввода): ")  
 if group == 'конец':  
 break  
  
 # Если группа уже есть в словаре, добавляем студента в эту группу  
 if group in students:  
 students[group].append(input("Введите имя студента: "))  
 # Иначе создаем новую группу и добавляем студента в нее  
 else:  
 students[group] = [input("Введите имя студента: ")]  
  
# Упорядочиваем список студентов по группам и выводим результат на экран  
for group, students\_list in sorted(students.items()):  
 print("Группа", group)  
 for student in sorted(students\_list):  
 print(" -", student)

2. Напишите программу которая возвращает список. Заранее подготовьте список предметов и оценок учащихся. Когда вы вводите имя учащегося, то должны отображаться оценки этого учащегося.

# Заранее подготовленный список предметов и оценок учащихся  
grades = {  
 'John': {  
 'Math': [5, 4, 3, 5],  
 'Science': [4, 5, 5, 4],  
 'History': [3, 4, 5, 3]  
 },  
 'Alice': {  
 'Math': [4, 5, 4, 5],  
 'Science': [5, 5, 4, 4],  
 'History': [4, 4, 3, 5]  
 },  
 'Bob': {  
 'Math': [3, 4, 5, 4],  
 'Science': [4, 4, 5, 5],  
 'History': [5, 3, 4, 4]  
 }  
}  
  
  
# Функция для получения оценок учащегося по имени  
def get\_grades(name):  
 if name in grades:  
 return grades[name]  
 else:  
 return None  
  
  
# Пример использования функции  
name = input("Введите имя учащегося: ")  
grades = get\_grades(name)  
print(type(grades))  
if grades is not None:  
 print("Оценки учащегося", name)  
 for subject, subject\_grades in grades.items():  
 print(subject, ':', subject\_grades)  
else:  
 print("Учащийся", name, "не найден")

3.Напишите программу, которая будет запрашивать у пользователя целочисленные значения и сохранять их в виде списка. Индикатором окончания ввода значений должен служить ноль. Затем программа должна вывести на экран все введенные пользователем числа (кроме нуля) в порядке возрастания–по одному значению в строке. Используйте для сортировки либо метод sort, либо функцию sorted.

# Создаем пустой список для хранения введенных чисел  
numbers = []  
  
# Запрашиваем у пользователя числа до тех пор, пока он не введет 0  
while True:  
 num = int(input("Введите число (для завершения введите 0): "))  
 if num == 0:  
 break  
 numbers.append(num)  
  
# Сортируем список чисел в порядке возрастания  
numbers.sort()  
  
# Выводим отсортированный список на экран  
print("Введенные числа в порядке возрастания:")  
for num in numbers:  
 print(num, end=" ")

4.Напишите программу, которая, как и в предыдущем случае, будет запрашивать у пользователя целые числа и сохранять их в виде списка. Индикатором окончания ввода значений также должен служить ноль. На этот раз необходимо вывести на экран введенные значения в порядке убывания.

# Создаем пустой список для хранения введенных чисел  
numbers = []  
  
# Запрашиваем у пользователя числа до тех пор, пока он не введет 0  
while True:  
 num = int(input("Введите число (для завершения введите 0): "))  
 if num == 0:  
 break  
 numbers.append(num)  
  
# Сортируем список чисел в порядке возрастания  
numbers.sort(reverse=True)  
  
# Выводим отсортированный список на экран  
print("Введенные числа в порядке возрастания:")  
for num in numbers:  
 print(num, end=" ")

5. Для выигрыша главного приза необходимо, чтобы шесть номеров на лотерейном билете совпали с шестью числами, выпавшими случайным образом в диапазоне от 1 до 49 во время очередного тиража. Напишите программу, которая будет случайным образом подбирать шесть номеров для вашего билета. Убедитесь в том, что среди этих чисел не будет дубликатов. Выведите номера билетов на экран по возрастанию.

import random  
  
numbers = sorted([random.randint(1, 49) for \_ in range(6)])  
numbers\_set = set(numbers)  
  
while len(numbers\_set) < 6:  
 numbers\_set.add(random.randint(1, 49))  
  
numbers = sorted(list(numbers\_set))  
  
print(numbers)

6. Напишите функцию, показывающую, отсортирован ли переданный ей в качестве параметра список (по возрастанию или убыванию). Функция должна возвращать True, если список отсортирован, и False в противном случае. В основной программе запросите у пользователя последовательность чисел для списка, после чего выведите сообщение о том, является ли этот список отсортированным изначально.

def is\_sorted(arr):  
 *"""  
 Функция, проверяющая отсортирован ли переданный список.  
 Возвращает True, если список отсортирован, и False в противном случае.  
 """* if arr == sorted(arr) or arr == sorted(arr, reverse=True):  
 return True  
 else:  
 return False  
  
  
# запрос последовательности чисел для списка  
arr = input("Введите последовательность чисел для списка, разделенных пробелами: ")  
arr = arr.split()  
arr = [int(x) for x in arr]  
  
# проверка, отсортирован ли список изначально  
if is\_sorted(arr):  
 print("Список отсортирован")  
else:  
 print("Список не отсортирован")