**Отчет по практической работе 7**

Выполнил обучающийся

ФИО: Цветков С.А.

**Задание 1.**

Составьте словарь для специалиста отдела кадров. Каждая строчка состоит из трёх частей, разделённых пробелом – фамилии сотрудника, дня и месяца его рождения. Фамилия – это строка из русских букв, день – число от 1 до 31, а месяц – строка из набора «январь», «февраль», «март», «апрель», «май», «июнь», «июль», «август», «сентябрь», «октябрь», «ноябрь», «декабрь». По известному списку всех дней рождения научитесь определять, у кого день рождения в заданном месяце.

Формат ввода

В первой строчке записано целое число N (1 ≤ N ≤ М), где М – количество сотрудников. В следующих N строчках записана информация об их днях рождения. Каждая строчка состоит из трёх частей, разделённых пробелом – фамилии сотрудника, дня и месяца его рождения. Фамилии всех сотрудников считаем различными. В следующей строчке записано целое число К (1 ≤ К ≤ 100) – количество вопросов, на которое надо ответить. В следующих К строках содержатся сами вопросы. Вопрос – это название месяца в том же формате, в котором они задаются выше.

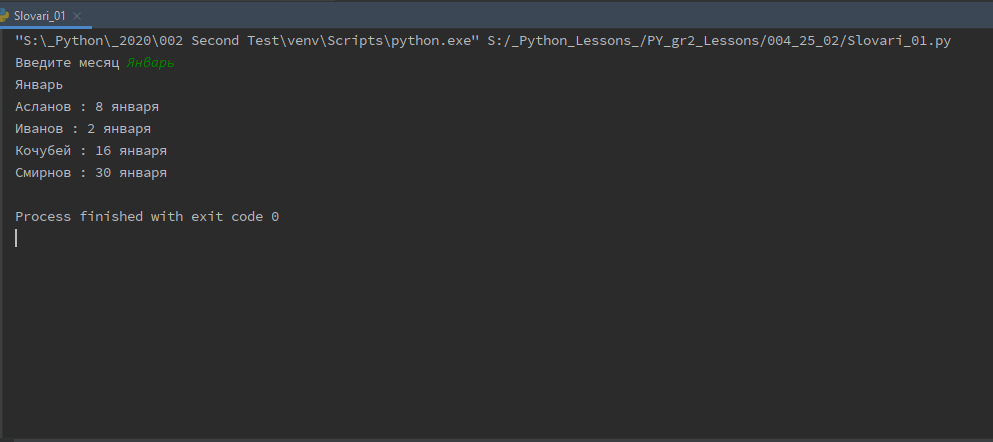
Формат вывода

Для каждого вопроса в отдельной строчке через пробел выведите фамилии всех сотрудников, которые родились в указанном месяце. Фамилии упорядочьте в лексикографическом порядке. Если в заданном месяце никто не родился, оставьте соответствующую строку ответа пустой.

Текст программы:

# Словари. Задание 1  
# Предположим, что у всех сотрудников - разные фамилии. В таком случае - их фамилии  
# можно использовать в качестве ключа  
# Задаем словарь сотрудников  
Empl = {  
 "Иванов":['2', 'январь'],  
 "Петров":['4', 'февраль'],  
 "Смирнов" : ['30' , 'январь'] ,  
 "Ворошилов" : ['28' , 'июнь'] ,  
 "Уткин" : ['12' , 'июль'] ,  
 "Карпин" : ['2' , 'февраль'] ,  
 "Махно" : ['31' , 'декабрь'] ,  
 "Кочубей" : ['16' , 'январь'] ,  
 "Закариадзе" : ['18' , 'октябрь'] ,  
 "Иванопуло" : ['1' , 'декабрь'] ,  
 "Катышев" : ['25' , 'февраль'] ,  
 "Лозовой" : ['8' , 'март'],  
 "Асланов" : ['8' , 'январь'],  
 "Кикабидзе" : ['8' , 'май'],  
 "Абакумов" : ['28' , 'февраль'],  
 "Цветков" : ['2' , 'февраль']  
}  
# Функция преобразования названия месяца  
def MonthToMon(Mon):  
 # Mon = Mon.lower()  
 if Mon == "январь": Month = "января"  
 if Mon == "февраль": Month = "февраля"  
 if Mon == "март": Month = "марта"  
 if Mon == "апрель": Month = "апреля"  
 if Mon == "май": Month = "мая"  
 if Mon == "июнь": Month = "июня"  
 if Mon == "июль": Month = "июля"  
 if Mon == "август": Month = "августа"  
 if Mon == "сентябрь": Month = "сентября"  
 if Mon == "октябрь": Month = "октября"  
 if Mon == "ноябрь": Month = "ноября"  
 if Mon == "декабрь": Month = "декабря"  
 return Month  
  
# Задаем словарь дней рождения в выбранном месяце  
EmplBirthday = {}  
BrdMonth = input("Введите месяц ")  
print(BrdMonth)  
for key, val in Empl.items():  
 if val[1] == BrdMonth.lower():  
 # Заполняем словарь выбранного месяца  
 EmplBirthday.update({ key:val[0]+" " + MonthToMon(val[1]) })  
  
# Вывод отсортированного списка ДР в выбранном месяце  
list\_key = list(EmplBirthday.keys())  
list\_key.sort()  
for i in list\_key:  
 print (i, ":", EmplBirthday[i])

Результат выполнения (скриншот):



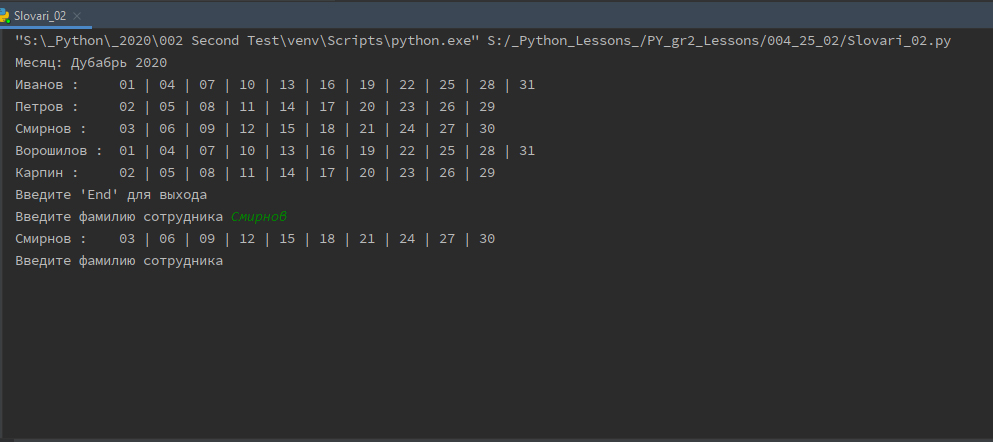
**Задание 2.**

Разработайте программу словаря, хранящего информацию о графике работы сотрудников в некотором месяце, при условии, что сотрудник работает «сутки через двое». Обеспечьте возможность получения информации о графике каждого сотрудника по вызову его фамилии.

Текст программы:

# Создаем и заполняем список дат месяца  
Month = ['{:02}'.format(i) for i in range(1,32)]  
MonthName = "Дубабрь 2020"  
# Задаем график работы для всех сотрудников [1 - каждый день, 2 - сутки сеиез сутки, 3 - сутки / двое]  
Schedule = 3  
# Формируем словарь  
Empl = {  
 "Иванов":[k for k in Month[::Schedule]],  
 "Петров":[k for k in Month[1::Schedule]],  
 "Смирнов":[k for k in Month[2::Schedule]],  
 "Ворошилов":[k for k in Month[::Schedule]],  
 "Карпин" : [k for k in Month[1: :Schedule]]  
}  
# Выводим общий гафик по компании  
print("Месяц: "+MonthName)  
for key, value in Empl.items():  
 print(key, ":\t", " | ".join(value))  
# Выводим график по отдельному сотруднику  
print("Введите 'End' для выхода")  
OneEmpl = ""  
while OneEmpl != "End":  
 OneEmpl = input("Введите фамилию сотрудника ")  
 if OneEmpl == 'End':  
 break  
 if OneEmpl in Empl:  
 print(OneEmpl, ":\t", " | ".join(Empl[OneEmpl]))  
 else:  
 print("Нет такого сотрудника")

Результат выполнения (скриншот):



**Задание 3.**

Дана база данных о продажах некоторого интернет-магазина. Каждая строка входного файла представляет собой запись вида «Покупатель товар количество», где Покупатель – имя покупателя (строка без пробелов), товар – название товара (строка без пробелов), количество – количество приобретенных единиц товара.

Создайте список всех покупателей, а для каждого покупателя подсчитайте количество приобретенных им единиц каждого вида товаров. Список покупателей, а также список товаров для каждого покупателя нужно выводить в лексикографическом порядке.

(\*Данное задание представлено на сайте «Питонтьютор». По ссылке https://pythontutor.ru/lessons/dicts/problems/sales/ Вы можете получить доступ к нему и проверить программу на представленных тестовых данных)

Текст программы:

# Импортируем модуль для работы с CSV - вместо пробела - используем ";"  
import csv  
# Читаем файл и помещаем его в список  
with open('product\_list.csv', 'r', encoding = 'UTF-8') as fp:  
 reader = csv.reader(fp, delimiter=';', quotechar='"')  
 data\_read = [row for row in reader]  
  
# Создаем новое множество покупателей (значения - уникальны)  
Buyer = set()  
for i in data\_read: Buyer.add(i[0])  
Buyer = sorted(list(Buyer)) # сортируем список покупателей по алфавиту  
  
# Выводим список покупателей и купленны х ими продукутов  
print("\*"\*40)  
for BuyerMame in Buyer:  
 print("\tПокупатель:",BuyerMame)  
 GoodsArr = {} # Создаем список продуктов  
 for Purchase in data\_read:  
 if Purchase[0] == BuyerMame:  
 Goods = Purchase[1]  
 GoodsNumber = int(Purchase[2])  
 # Заполняем список  
 if not Goods in GoodsArr:  
 GoodsArr.update( {Goods:GoodsNumber} )  
 else:  
 GoodsNumber = int(GoodsArr[Goods]) + int(Purchase[2])  
 GoodsArr.update ( {Goods : GoodsNumber} )  
  
 # Здесь выводим отсортированный по ключу список товаров для каждого покупателя  
 list\_key = list ( GoodsArr.keys () )  
 list\_key.sort ()  
 for i in list\_key :  
 print ( "\t", i , ":" , GoodsArr[i] )  
 print("\*"\*40)

Результат выполнения (скриншот):

