***Лабораторна робота № 3***

***Тема:*** Списки. Кортежі. Сети. Словники.

***Мета:*** набути практичних навичок створення програм, що використовують зазначені конструкції, їх написання та налагодження на мові програмування Python, вивчення основних конструкцій, які використовуються в подібних програмах.

***Короткі теоретичні відомості***

Основні конструкції на мові Python:

1. Список (list) – це змінна впорядкована послідовність елементів. Елементи списку також розділяються комами, але задаються вже в квадратних дужках. Для створення списків використовується спеціальна вбудована функція list().
2. Кортеж (tuple) являє собою незмінну впорядковану послідовність даних. У ньому можуть міститися елементи різних типів, наприклад інші кортежі. Кортеж визначається в круглих дужках, а його елементи розділяються комами. Спеціальна вбудована функція tuple() дозволяє створювати кортежі з представленої послідовності даних.
3. Сет або множина (Set) являє собою невідсортовану колекцію унікальних елементів. Сет визначається в фігурних дужках. Сети не підтримують індексацію і зрізи.
4. Словник (dictionary) є хеш-таблицею, що зберігає елемент разом з його ідентифікатором-ключем. Наступний доступ до елементів виконується теж по ключу, тому одиниця зберігання в словнику – це пара об'єкт-ключ і пов'язаний з ним об'єкт-значення. Словник – це змінна, але не впорядкована колекція, так що порядок елементів в словнику може змінюватися з часом. Здається словник в фігурних дужках, ключ відділяється від значення двокрапкою, а самі пари ключ-значення розділяються комами. Для створення словників використовується спеціальна вбудована функція dict().

***Результати роботи:***

***Завдання 1*** – Извлечение и удаление элементов списка.

е) Создать список, состоящий из целых чисел указанного диапазона. Удалить из каждого списка первый элемент все последующие со значением кратным 10 и найти их количество.

Код програми:

#Ihor Mostovyi

#20.04.2020

from datetime import datetime

def main():

start = (int(input("Enter start of range\n")))

end = (int(input("Enter end of range\n"))) + 1

if end < start:

temp = start

start = end

end = temp

myList = list(range(start, end))

del myList[0]

result = list(filter(lambda i: i % 10 != 0, myList))

count = len(result)

print(result, "\n")

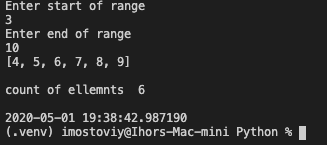
print("count of ellemnts ", count, "\n")

print(datetime.now())

return None

main()

Результат роботи:



***Завдання 2*** – Сгенерировать два списка чисел со значениями из заданного диапазона, которые могут содержать до 100000 чисел каждый. Посчитайте, сколько чисел:

e)  нечетных, повторяющихся в обоих списках, а также их количество

Код програми:

#Ihor Mostovyi

#20.04.2020

import collections

from datetime import datetime

from Helpers import getStartRange, makeRandomList

def calculate(listTocount: list):

dictionary = {}

for item, count in collections.Counter(listTocount).items():

if (count > 1 and item % 2 != 0):

dictionary[item] = count

return dictionary

def main():

rangeToGetForm = getStartRange()

dictOne = calculate(makeRandomList(rangeToGetForm))

dictTwo = calculate(makeRandomList(rangeToGetForm))

print("count of digits that satisfy the condition if first list \n", dictOne, "\n and in second\n", dictTwo)

print("\n", datetime.now())

return None

main()

import random

def makeRandomList(rangeToGetForm: range):

n = random.randint(1, 10000)

randomList = []

for \_ in range(0, n + 1):

rand = random.choice(rangeToGetForm)

randomList.append(rand)

return randomList

def getStartRange():

start = (int(input("Enter start of range\n")))

end = (int(input("Enter end of range\n")))

if end < start:

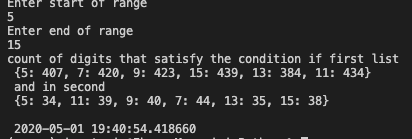
temp = start

start = end

end = temp

return range(start, end + 1)

Результат роботи:



***Завдання 3*** – Даны два списка чисел, которые могут содержать до 10000 чисел каждый. Выведите все числа, которые входят как в первый, так и во второй список в порядке

e)  возрастания и все кратны 3;

Код програми:

# Ihor Mostoviy

# 20.04.2020

import random

from datetime import datetime

from Helpers import getStartRange, makeRandomList

def main():

ragneToGetFrom = getStartRange()

setOne = set(makeRandomList(ragneToGetFrom))

setTwo = set(makeRandomList(ragneToGetFrom))

intersection = list(setOne.intersection(setTwo))

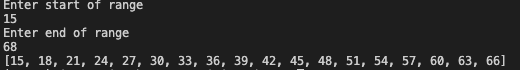
intersection.sort()

print(list(filter(lambda x: x % 3 == 0, intersection)))

return None

main()

Результат роботи:



***Завдання 4*** – Анна и Борис любят играть в разноцветные кубики, причем у каждого из них свой набор и в каждом наборе все кубики различны по цвету. Однажды дети заинтересовались, сколько существуют цветов таких, что кубики каждого цвета присутствуют в обоих наборах. Для этого они занумеровали все цвета случайными числами. На этом их энтузиазм иссяк, поэтому вам предлагается помочь им в оставшейся части. Номер любого цвета – это целое число в пределах  
от 0 до 109 . В первой строке входного файла записаны числа N и M – количество кубиков у Анны и Бориса соответственно. В следующих N строках заданы номера цветов кубиков Анны. В последних M строках номера цветов кубиков Бориса.

e)  Выведите сначала количество, а затем отсортированные по возрастанию номера цветов таких, что кубики каждого цвета есть в обоих наборах, затем количество и отсортированные по убыванию кратные 5 номера остальных цветов у Анны, потом количество и отсортированные по возрастанию кратные 3 номера остальных цветов у Бориса.

Код програми:

def getNumbers():

n = int(input("count of Ann's cubes\n"))

m = int(input("count of Borise's cubes\n"))

endOfRange = 10 \*\* 9

annsList = set()

borisList = set()

while len(annsList) != n:

digit = int(input("enter number of color for Ann\n"))

if digit >= 0 and digit < endOfRange:

annsList.add(digit)

else:

print("Enter number from 0 to 10^9\n")

while len(borisList) != m:

digit = int(input("enter number of color for Boris\n"))

if digit >= 0 and digit < endOfRange:

borisList.add(digit)

else:

print("Enter number from 0 to 10^9\n")

return annsList, borisList

def main():

annsList, borisList = getNumbers()

intersectionList = list(annsList.intersection(borisList))

intersectionList.sort()

print("intersection colors count is", len(intersectionList), "\n colors are\n", intersectionList, "\n")

leftForAnn = list(annsList.difference(set(intersectionList)))

leftForAnn.sort(reverse = True)

result = list(filter(lambda x: x % 5 == 0, leftForAnn))

print("Left for ann with condition\n", result, "\n")

leftForBoris = list(borisList.difference(set(intersectionList)))

leftForBoris.sort()

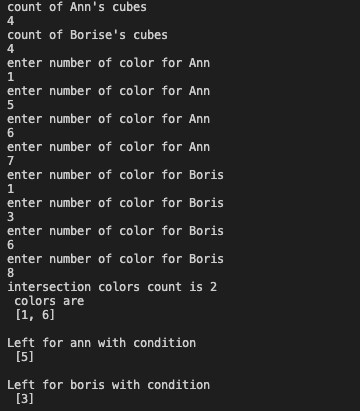
result = list(filter(lambda x: x % 3 == 0, leftForBoris))

print("Left for boris with condition\n", result, "\n")

return None

main()

Результат роботи:



***Завдання 5*** – Анна и Николай играют в игру. Август загадал натуральное число от 1 до n (задается пользователем). Анна пытается угадать это число, для этого она называет некоторые множества натуральных чисел. Николай отвечает Анне YES, если среди названных ей чисел есть задуманное или NO – в противном случае. После нескольких заданных вопросов Анна запуталась в том, какие вопросы она задавала и какие ответы получила и просит вас помочь ей определить, какие числа мог задумать Николай.

e)  Первая строка входных данных содержит число 2n – наименьшее число, которое мог загадать Николай. Далее идут строки, содержащие вопросы Анны. Каждая строка представляет собой набор чисел, разделенных символами подчеркивания. После каждой строки с вопросом идет ответ Николая: YES или NO. Наконец, последняя строка входных данных содержит одно слово HELP. Вы должны вывести (через апостроф, в порядке возрастания) все числа кратные 11, которые мог задумать Николай.

Код програми:

#Ihor Mostovyi

#20.04.2020

def testString(n, string):

splitedList = string.split("\_")

if str(n) in splitedList:

print("YES\n")

return None

print("NO\n")

return None

def main():

n = int((int(input("Enter 2\*n\n"))) / 2)

string = input("Enter sequence devided buy '\_'")

while string != "HELP":

testString(n, string)

string = input("Enter sequence devided buy '\_'")

listOfDigits = list(filter(lambda x: x % 11 == 0, list(range(1, n + 1))))

print("Possible digits are: \n")

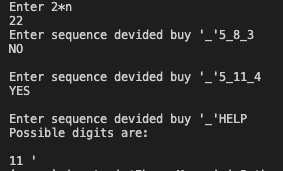
for i in listOfDigits:

print(i, "' ")

return None

main()

Результат роботи:



1. ***Завдання 6*** – Политическая жизнь некоторой страны очень оживленная. В стране действует K политических партий, каждая из которых регулярно проводит некие политические акции. Дни, когда хотя бы одна из партий объявляет политические акции, при условии, что это не суббота или воскресенье (когда и так никто не работает), наносят большой ущерб экономике страны. i-я партия объявляет политические акции строго каждые bi дней, начиная с дня с номером ai . То есть i-я партия объявляет политические акции в дни ai , ai + bi , ai + 2bi и т.д. Если в какой-то день несколько партий объявляет политические акции, то это считается одной общенациональной политической акцией. В календаре страны в 2017 году n дней, пронумерованных от 1 до N. Первый день года является понедельником, шестой и седьмой дни недели – выходные. Программа получает на вход число дней в году n и число политических партий 1 ≤ k ≤ 250 . Далее идет K строк, описывающие графики проведения политических акций. i-я строка содержит числа ai и bi ( 1 ≤ ai , bi ≤ n ). Выведите единственное число: количество политических акций, произошедших в течение года.

Код програми:

# Ihor Mostovyi

# 20.04.2020

def main():

k = int(input("Enter count of clubs\n"))

n = int(input("Enter count of days\n"))

clubs = []

while len(clubs) != k:

string = input("Enter ai and bi, 1<=ai, bi<=n\n")

numbers = string.split(" ", 1)

a = int(numbers[0])

b = int(numbers[1])

if a < 1 or b > n:

print("invalid string\n")

continue

days = set(filter(lambda x: x % 6 != 0 and x % 7 != 0, list(range(int(numbers[0]), n + 1, int(numbers[1])))))

clubs.append(days)

greatDays = clubs[0]

for days in clubs:

greatDays = greatDays.intersection(days)

daysWithoutGreatDays = list(map(lambda x: x.difference(greatDays), clubs))

result = len(greatDays)

for days in daysWithoutGreatDays:

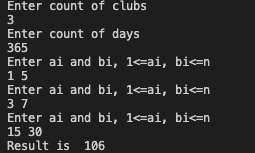
result += len(days)

print("Result is ", result)

return None

main()

Результат роботи:



1. ***Завдання 7*** – Дан список слов (пользователь вводит фразу, содержащую не менее 15 слов) x1,Κ,xn и слово y (ключевое слово выбирается указанием его индекса во фразе). Определить входит ли хотя бы одно из слов xi в y (как подслово). Количество действий не должно превосходить константы согласно варианту, умноженной на суммарную длину всех слов (из основного списка и того в котором происходит поиск).

e) K=1/2

Код програми:

# Ihor Mostovyi

# 20.04.2020

def calculateConst(string):

listOfWords = string.split(" ")

lengthOfWords = 0

for word in listOfWords:

lengthOfWords += len(word)

return lengthOfWords

def main():

string = input("Enter phrases, splited by whitespaces\n")

listOfWords = string.split(" ")

k = calculateConst(string)

k1 = 1

inp = "Enter key word(you nead to choose index between 0 and " + str(len(listOfWords) - 1) + "\n"

n = int(input(inp))

if n < 0 or n >= len(listOfWords):

print("Did u read condition??!")

return None

keyword = listOfWords[n]

keywordLen = len(keyword)

contain = False

for word in listOfWords:

if word == keyword:

k1 += 1

continue

if len(word) < keywordLen:

k1 += 1

continue

if keyword in word:

k1 += 1

contain = True

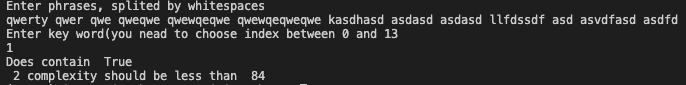
break

print("Does contain ", contain, "\n", k1, "complexity should be less than ", k)

return None

main()

Результат роботи:



***Завдання 8*** – Стеки і черги:

1. Создать стек, информационными полями которого являются: улица, номер дома и номер квартиры. Добавить в стек сведения о новой квартире. Организовать просмотр данных стека, предварительно удалив из него все адреса с нечетными номерами домов, и определить количество домов на заданной улице.

Код програми:

#Ihor Mostovyi

#20.04.2020

def main():

stack = [["Street 1", 12, 3],

["Street 1", 13, 4],

["Street 2", 2, 3],

["Street 3", 126, 13],

["Street 3", 11, 33],

["Street 3", 121, 13],

["Street 3", 141, 86]]

print("Stack at start\n", stack )

print("Add new flat\n")

street = input("Enter street ")

house = int(input("Enter house number "))

flat = int(input("Enter flat number "))

stack.insert(0, [street, house, flat])

print("Stack state \n", stack)

filtred = list(filter(lambda x: x[1] % 2 == 0, stack))

print("Stack lookup\n", filtred)

filtred2 = list(filter(lambda x: x[0] == "Street 3", stack))

houses = set()

for adress in filtred2:

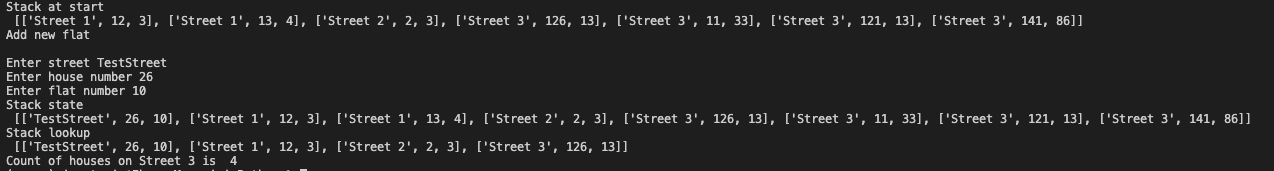
houses.add(adress[1])

print("Count of houses on Street 3 is ", len(houses))

return None

main()

Результат роботи (рис. 18-19):



***Завдання 9*** – Список содержит названия городов Украины, области к которым они относятся, количество жителей в них и общую площадь территории. На основе данного создать списки, которые: разбивают текущий на группы: в одном – города милионники, в другом – остальные;

Код програми:

# Ihor Mostovyi

# 20.04.2020

def main():

regions = [["Vinitsya", "Vinnitska", 1575808, 26513],

["Lytsk", "Volunska", 1038457, 20143],

["Dnipro", "Dnipropetrovska", 3231140, 31914],

["Donetsk", "Donetska", 4200461, 26517],

["Shitomur", "Shitomurska", 1231239, 29832],

["Ushgorod", "Zakarpatska", 1258115, 12777],

["Zaporishya", "Zaporizka", 1723171, 27180],

["Ivano-Frankivsk", "Ivano-Frankivska", 1377496, 13928],

["Kyiv", "Kyivska", 1754284, 28131],

["Kropuvnuckui", "Kirivigradska", 956250, 24588],

["Luhansk", "Luhanska", 2167802, 26684],

["Lviv", "Lvivska", 2529608, 21883],

["Mukolaiv", "Mukolaivska", 1141324, 24598],

["Odesa", "Odeska", 2383075, 33310],

["Poltava", "Poltavska", 1466786, 28748],

["Rivne", "Rivnenka", 1160647, 20047],

["Symu", "Symska", 1094284, 23834],

["Ternopil", "Ternopilska", 1052312, 13823],

["Harkiv", "harkivska", 2694007, 31415],

["Herson", "Hersonska", 1046981, 28461],

["Kamyanec-Podilskuy", "Hmelnucka", 1274409, 20629],

["Cherkasy", "Cherkaska", 1220364, 20900],

["Chernivci", "Chernivecka", 906600, 8097],

["Chernigiv", "Chernivecka", 1020078, 31865]]

regionsCopy = [region for region in regions]

regions.sort(key = lambda x: x[1])

regionsCopy.sort(key = lambda x: x[2])

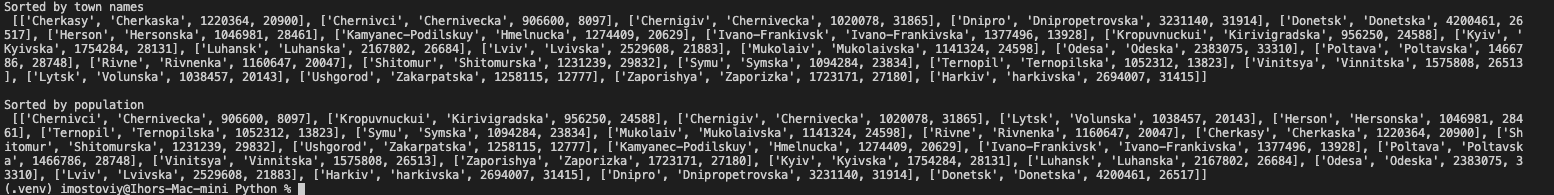
print("Sorted by town names\n", regions, "\n")

print("Sorted by population\n", regionsCopy)

return None

main()

Результат роботи:



***Завдання 10*** – Создать словарь-разговорник по указанной в варианте теме, содержащий 10 слов вида: слово на украинском и его перевод. Вывести на экран измененный словарь, в котором первым идет слово на инострранном из входного словаря, а рядом его украинский перевод.

e)  украинско-польский, тема словаря – здоровье

Код програми:

# Ihor Mostovyi

# 20.04.2020

def main():

ucrainianPolandDictionaty = {"Здоровий": "Zdrowe",

"Зарядка": "Ładowanie",

"Хвороба": "Choroba",

"Спорт": "Sports",

"Тренування": "Szkolenie",

"Грип": "Grypa",

"М'язи": "Mięśnie",

"Жир": "Tłuszcz",

"Каріес": "Próchnica",

"Сильний": "Silny"}

poland\_ukrainian\_dictionary = dict()

for key, value in ucrainianPolandDictionaty.items():

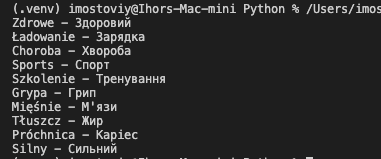
poland\_ukrainian\_dictionary[value] = key

for key, value in poland\_ukrainian\_dictionary.items():

print(key, '-', value)

main()

Результат роботи:



***Висновки:*** Було вивчено теоретичні відомості з теми «Списки. Кортежі. Сети. Словники». Було набуто практичних навичок створення програм, що використовують зазначені конструкції, їх написання та налагодження на мові програмування Python, вивчення основних конструкцій, які використовуються в подібних програмах.