***Лабораторна робота № 5***

***Тема:*** Одномірні і багатовимірні масиви в Python. Знайомство з NumPy: створення масивів, застосування базових операцій і функцій. Робота з файлами.

***Мета:*** набуття практичних навичок створення програм з використанням розширень мови Python.

***Короткі теоретичні відомості***

Клас array стандартної бібліотеки Python використовується тільки для одновимірних масивів. З основними методами по роботі з класом array можна ознайомитися через help ( 'array').

NumPy – це розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами, тому використовується частіше.

Крім NumPy роботу з масивами даних підтримує відкрита бібліотека SciPy. Крім того, SciPy містить модулі для оптимізації, інтегрування, спеціальних функцій, обробки сигналів, обробки зображень, генетичних алгоритмів, рішення звичайних диференціальних рівнянь та інших задач, зазвичай вирішуються в науці і при інженерної розробці. Бібліотека розробляється для тієї ж аудиторії, що MATLAB і Scilab. Для візуалізації при використанні SciPy часто застосовують бібліотеку Matplotlib, що є аналогом засобів виведення графіки MATLAB. В даний час SciPy поширюється під ліцензією BSD і його розробники спонсоруються Enthought.

Можливості:

* пошук мінімумів і максимумів функцій;
* обчислення інтегралів функцій;
* підтримка спеціальних функцій;
* обробка сигналів;
* обробка зображень;
* робота з генетичними алгоритмами;
* рішення звичайних диференціальних рівнянь;
* інші можливості.

***Результати роботи:***

***Завдання 1*** – Сохраните текстовый файл, содержащий несколько строк стихотворения украинского поэта. Считайте данные из файла в строковый массив и организуйте вывод данных на экран. Подсчитайте количество слов и букв в словах, найдите слово максимальной и минимальной длины, отсортируйте массив по возрастанию и сохраните данные в файл. Создайте одномерный массив из значений, соответствующих количествам букв в словах и вычислите, используя функции NumPy:

e)  среднее гармоническое элементов данного массива и логарифма его максимального элемента;

Код програми:

# Ihor Mostovyi

# 29.04.2020

import numpy as np

from math import log10 as log

import os, sys

def main():

input\_file = open(os.path.join(sys.path[0], 'text1.txt'), 'r')

lines = []

for line in input\_file:

if (line == "\n"):

continue

str\_arr = line.split(" ")

for j in range(0, len(str\_arr), 1):

lines.append(str\_arr[j].rstrip())

input\_file.close()

words = np.array([word for word in lines], str)

words = sorted(words, key=lambda x: len(x), reverse=False)

wordsLengths = np.array([len(word) for word in words], int)

minLengthWord = words[0]

maxLengthWord = words[len(words) - 1]

print("Count of words: " + str(len(words)), "\n")

print("Word with min len (", len(minLengthWord), "): ", minLengthWord)

print("\nWord with max len (", len(maxLengthWord), "): ", maxLengthWord)

harmonicMean = len(wordsLengths) / np.sum(1.0/wordsLengths)

print("Harmonic mean is: ", harmonicMean)

print("Log of max length word is: Log(", len(maxLengthWord), ") = ", log(len(maxLengthWord)))

output\_file = open(os.path.join(sys.path[0], 'task1\_result.txt'), 'w')

output\_file.write("List of words: " + str(lines) + "\n")

output\_file.write("Count of words: " + str(len(words)) + "\n")

output\_file.write("Count of letters in each word: " + str(len(words)) + "\n")

a = np.array([len(word) for word in lines], int)

output\_file.write("Initial array: " + str(a) + "\n")

output\_file.write("Sorted array: " + str(wordsLengths) + "\n")

Sum = a.sum()

print("Sum of elements of array = ", Sum)

output\_file.close()

return None

main()

Файл з вхідними даними:

Мені тринадцятий минало.

Я пас ягнята за селом.

Чи то так сонечко сіяло,

Чи так мені чого було?

Мені так любо, любо стало,

Неначе в Бога......

Уже прокликали до паю,

А я собі у бур’яні

Молюся Богу... І не знаю,

Чого маленькому мені

Тойді так приязно молилось,

Чого так весело було.

Господнє небо, і село,

Ягня, здається, веселилось!

І сонце гріло, не пекло!

Та недовго сонце гріло,

Недовго молилось...

Запекло, почервоніло

І рай запалило.

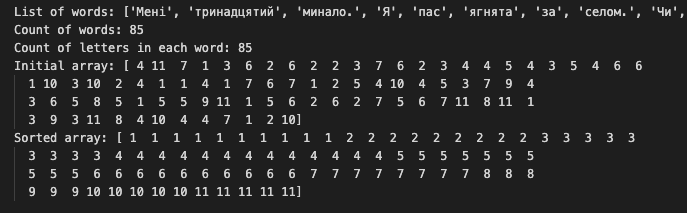
Мов прокинувся, дивлюся:

Село почорніло,

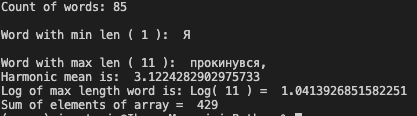
Боже небо голубеє

І те помарніло.

Файл з вихідними даними:



Результат роботи:



***Завдання 2*** – Заданы 2 одномерных массива состоящие из n целых чисел, которые заполняются случайным образом. Составить программу формирования третьего массива:

e)  включая в него сначала все элементы первого, затем второго массивов;

Код програми:

#Ihor Mostovyi

#29.04.2020

import numpy as np

def main():

n = int(input("Enter the count of elements in array n: "))

firstArrray = np.random.randint(0, 100, n)

secondArrray = np.random.randint(0, 100, n)

print("First array: ", firstArrray, "\n")

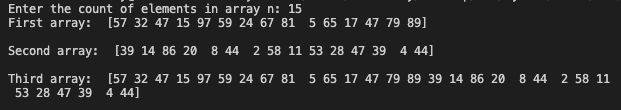
print("Second array: ", secondArrray, "\n")

print("Third array: ", np.concatenate((firstArrray, secondArrray), axis=0), "\n")

return None

main()

Результат роботи:



***Завдання 3*** – Составить программу вывода на экран:

e)  первых 100 нечетных натуральных нечетных чисел и распечатать его в виде матрицы 10х10;

Код програми:

#Ihor Mostovyi

#29.04.2020

import numpy as np

def getMatrix():

array = np.array([i for i in range(1, 200, 2)], int)

array = array.reshape((10, 10))

return array

def printing():

matrix = getMatrix()

print(len(matrix)\*len(matrix[0]))

print(matrix)

return None

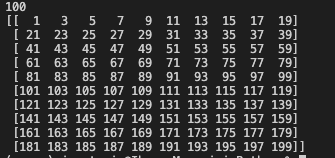
def main():

printing()

return None

main()

Результат роботи:



***Завдання 4*** – Усовершенствуйте программу (задача 6) таким образом, чтобы создать класс Determinant состоящий из методов:

a)  main – содержит код задачи 6, в котором полученная матрица обрезана до размера 3х3, а также обеспечивается ее вывод на экран в табличном формате путем вызова метода PrintMatrix и производится вычисление детерминанта матрицы путем вызова метода det;

b)  det – статический метод, возвращающий для переданной ему в качестве параметра матрицы значение ее детерминанта, вычисленного по правилу треугольника;

c)  PrintMatrix – производит вывод на экран матрицы переданной ему в качестве параметра в форме таблицы 3х3.

Код програми:

#Ihor Mostovyi

#29.04.2020

import numpy as np

import Task3 as t3

class Determinant:

def MainMethod(self):

matrix = t3.getMatrix()

matrix = matrix[:3, :3]

self.PrintMatrix(matrix)

det = self.det(matrix)

print(det)

self.check(det, matrix)

return None

def PrintMatrix(self, matrix):

print(matrix)

return None

def det(self, matrix):

det = 0

det += matrix[0, 0] \* matrix[1, 1] \* matrix[2, 2]

det += matrix[2, 0] \* matrix[0, 1] \* matrix[1, 2]

det += matrix[1, 0] \* matrix[0, 2] \* matrix[2, 1]

det -= matrix[2, 0] \* matrix[1, 1] \* matrix[0, 2]

det -= matrix[1, 0] \* matrix[0, 1] \* matrix[2, 2]

det -= matrix[0, 0] \* matrix[2, 1] \* matrix[1, 2]

return det

def check(self, det, matrix):

library\_value = np.linalg.det(matrix)

print("Library value of determinant = ", str(library\_value))

if (abs(det - library\_value) <= 0.0000001):

print("U deam good, boy")

else:

print("Go and read Math books!")

return None

def main():

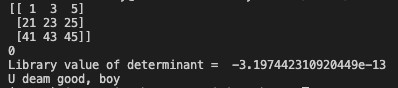
detMatrix = Determinant()

detMatrix.MainMethod()

return None

main()

Результат роботи:



***Завдання 5*** – Усовершенствуйте программу (задача 6) таким образом, чтобы создать класс SumMatrix состоящий из методов:

a)  main – содержит код задачи 6, в котором полученная матрица обрезана до размера 3х3, а также создается второй двумерный массив (матрица размером 3х3), который заполняется случайными числами в диапазоне от 0 до 5, затем с помощью метода SumMatrix производится их сложение и с помощью метода det вычисляется детерминант результирующей матрицы. Каждая операция должна сопровождаться выводом на экран матриц посредством вызова метода PrintMatrix;

b)  SumMatrix – производит сложение матриц переданных ему в качестве параметров.

Код програми:

#Ihor Mostovyi

#29.04.2020

import numpy as np

from Task3 import getMatrix

class SumMatrix:

def MainMethod(self):

m1 = getMatrix()

m1 = m1[:3, :3]

m2 = np.random.randint(0, 5, (3, 3))

self.PrintMatrix(m1)

self.PrintMatrix(m2)

sumOfMatrix = self.SumMatrix(m1, m2)

self.PrintMatrix(sumOfMatrix)

self.det(sumOfMatrix)

return None

def PrintMatrix(self, matrix):

print(matrix)

return None

def SumMatrix(self, a, b):

result = a + b

return result

def det(self, matrix):

print("determinant: ", np.linalg.det(matrix))

return None

def main():

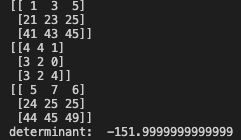
sumMatrix = SumMatrix()

sumMatrix.MainMethod()

return None

main()

Результат роботи:



***Завдання 6*** – Усовершенствуйте программу (задача 7) таким образом, чтобы создать класс Expression состоящий из методов:

a) main – содержит код задачи 6, в котором полученная матрица обрезана до размера 3х3, а также создается второй двумерный массив (матрица размером 3х3), который заполняется случайными числами в диапазоне от 0 до 5, затем с помощью метода ExpressionMatrix производится их сложение по формуле и с помощью метода det вычисляется детерминант результирующей матрицы. Каждая операция должна сопровождаться выводом на экран матриц посредством вызова метода PrintMatrix.  
b) ExpressionMatrix – производит вычисление 3\*A+5\*B.

Код програми :

#Ihor Mosovyi

#29.04.2020

import numpy as np

from Task3 import getMatrix

class SumMatrix:

def MainMethod(self):

m1 = getMatrix()

m1 = m1[:3, :3]

m2 = np.random.randint(0, 5, (3, 3))

self.PrintMatrix(m1)

self.PrintMatrix(m2)

result = self.ExpressionMatrix(m1, m2)

self.PrintMatrix(result)

print("determinant: ", self.det(result))

return None

def PrintMatrix(self, matrix):

print(matrix, "\n")

return None

def ExpressionMatrix(self, a, b):

a = 3\*a

b = 5\*b

self.PrintMatrix(a)

self.PrintMatrix(b)

return a + b

def det(self, matrix):

return np.linalg.det(matrix)

def main():

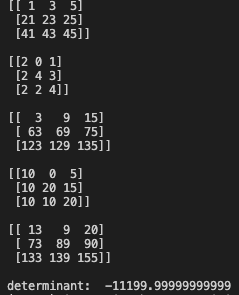
sumMatrix = SumMatrix()

sumMatrix.MainMethod()

return None

main()

Результат роботи:



***Завдання 7*** – Создайте класс SumElements состоящий из методов:  
a) main - в котором создается два двумерных массива (матрицы размером 5х10), которые заполняются случайными числами из разных диапазонов, после чего создаются еще две матрицы:  
− матрица-столбец, каждый элемент которой является суммой элементов исходной матрицы в соответствующей строке;  
− матрица-строка, каждый элемент которой является суммой элементов исходной матрицы в соответствующем столбце.  
b) PrintMatrix - производит вывод на экран матрицы переданной ему в качестве параметра в форме таблицы.

Код програми:

# Ihor Mostovyi

# 29.04.2020

import numpy as np

class SumElements:

def main(self):

start = int(input("Enter the start of diapason 1: "))

end = int(input("Enter the end of diapason 1: "))

matrixOne = np.random.randint(start, end, (5, 10))

start = int(input("Enter the start of diapason 2: "))

end = int(input("Enter the end of diapason 2: "))

matrixTwo = np.random.randint(start, end, (5, 10))

self.PrintMatrix(matrixOne)

self.PrintMatrix(matrixTwo)

matrixSumOfRows = matrixOne.sum(axis=1).reshape(5, 1)

matrixSumOfCols = matrixTwo.sum(axis=0).reshape(1, 10)

self.PrintMatrix(matrixSumOfRows)

self.PrintMatrix(matrixSumOfCols)

def PrintMatrix(self, matrix):

print(matrix, "\n")

return None

def main():

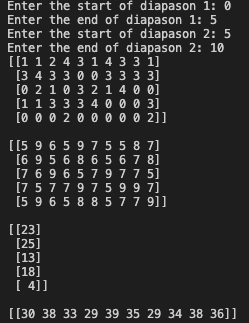
sumElements1 = SumElements()

sumElements1.main()

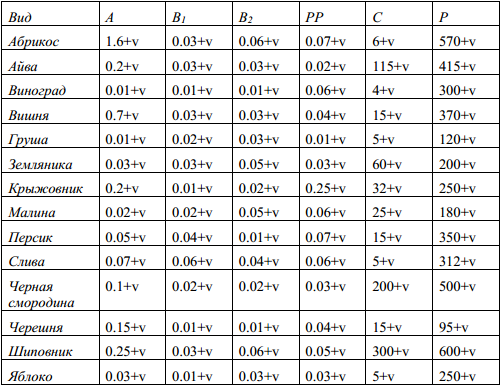
return None

main()

Результат роботи:



***Завдання 8*** – Имеется таблица данных о витаминном составе плодов и ягод:



Скласти програму обробки цих даних, що дозволяє визначити:

a) в каком из фруктов содержание заданного витамина максимально;  
b) массу каждого фрукта, необходимую для получения определенного количества заданного витамина;  
c) количественное содержание витаминов в заданном наборе определенных фруктов.

Код програми:

#Ihor Mostovyi

#29.04.2020

import numpy as np

import os, sys

def main():

input\_file = open(os.path.join(sys.path[0], "text2.txt"), 'r')

fruits = dict()

vitamins = list()

for line in input\_file:

if (line.\_\_contains\_\_("Вид")):

vit = line.split()

for i in range(1, len(vit) - 1, 1):

vitamins.append(vit[i])

continue

strs = line.split()

fruit = strs[0]

vitaminsValues = [float(i) for i in strs[1:len(strs)]]

fruits[fruit] = vitaminsValues

input\_file.close()

print("All vitamins: " ,vitamins)

for key, value in fruits.items():

print(key, " - ", value)

search\_vitamin = str(input("\nEnter a vitamin: "))

indexOfEnteredVitamin = vitamins.index(search\_vitamin)

maxvalueOfVitamin = 0

fruit = ""

for key, value in fruits.items():

if (value[indexOfEnteredVitamin] > maxvalueOfVitamin):

maxvalueOfVitamin = value[indexOfEnteredVitamin]

fruit = key

print("\nFruit: ", fruit, ", weight of vitamin = ", maxvalueOfVitamin, "\n")

requestedWeightOfVitamin = float(input("Enter count of vitamine: "))

for key, value in fruits.items():

count = value[indexOfEnteredVitamin]

weight = value[len(value)-1]

print("Fruit: ", key, ", weight = ", (requestedWeightOfVitamin \* weight) / count)

fruitsList = input("\nEnter fruits, separator mark is ',' ").split(",")

vitaminsCount = [0 for i in range(0, len(vitamins), 1)]

for key, value in fruits.items():

if (fruitsList.\_\_contains\_\_(key)):

for i in range(0, len(value) - 1, 1):

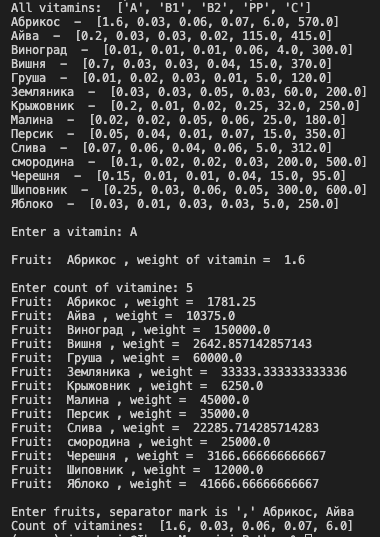
vitaminsCount[i] += value[i]

print("Count of vitamines: ", vitaminsCount)

return None

main()

Результат роботи:



***Завдання 9*** – Для ведения рейтинга группы сформировать таблицу, содержащую список студентов, список предметов одного семестра и результирующий рейтинг по ним (использовать данные прошлого семестра – результаты сессии/аттестации). Составить программу обработки этих данных, позволяющую:

a) вычислить средний рейтинг группы по всем предметам;

b)  вычислить средний рейтинг группы по каждому из предметов;

c)  вывести на экран фамилии студентов с наибольшим рейтингом;

d)  вывести на экран фамилии студентов с наименьшим рейтингом.

Результаты сохранить в текстовый файл.

Код програми:

#Ihor Mostovyi

#29.04.2020

import numpy as np

import operator

import os, sys

def main():

inputFile = open(os.path.join(sys.path[0], "text3.txt"), 'r')

students = dict()

subjects = dict()

for line in inputFile:

if (line.\_\_contains\_\_("ПІБ")):

subj = line.split()

for i in range(1, len(subj), 1):

subjects[subj[i]] = 0

continue

strs = line.split()

student = strs[0]

marks = [int(i) for i in strs[1:len(strs)]]

students[student] = marks

inputFile.close()

subjectIndex = 0

for key1, \_ in subjects.items():

marksSum = 0

for \_, value2 in students.items():

marksSum += value2[subjectIndex]

subjectIndex += 1

subjects[key1] = (marksSum / len(students))

outputFile = open(os.path.join(sys.path[0], 'task2\_result.txt'), 'w')

print("Average rating of subjects:")

outputFile.writelines("Average rating of subjects:")

for key, value in subjects.items():

print(str(key) + " - " + str(value))

outputFile.writelines(str(key) + " - " + str(value))

rating = dict()

for key, value in students.items():

MarksSum = 0

for mark in value:

MarksSum += mark

rating[key] = (MarksSum / len(value))

print()

outputFile.write("\n")

print("Average rating of group:")

outputFile.writelines("Average rating of group:")

for key, value in rating.items():

print(str(key) + " - " + str(value))

outputFile.writelines(str(key) + " - " + str(value))

print()

outputFile.write("\n")

rating = sorted(rating.items(), key=operator.itemgetter(1))

maxRating = rating[len(rating)-1]

print("Max rating:", maxRating)

outputFile.writelines("Max rating:")

for rate in rating:

if (rate[1] == maxRating[1]):

print(rate)

outputFile.writelines(str(rate) + " ")

print()

outputFile.write("\n")

minRating = rating[0]

print("Min rating:")

outputFile.writelines("Min rating:")

for rating in rating:

if (rating[1] == minRating[1]):

print(rating)

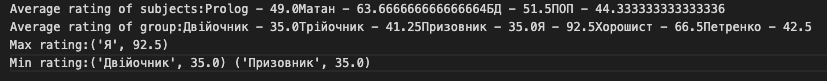
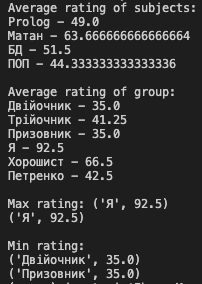
outputFile.writelines(str(rating) + " ")

outputFile.close()

return None

main()

Результат роботи:



***Завдання 10*** Создайте текстовый файл, содержащий следующие данные: названия городов для заданных в варианте областей Украины, численность населения городов, площадь городов, количество больниц и ВУЗов. Составить программу, которая сохраняет в строки текстового файла отсортированные по убыванию области в зависимости от:

a)  1 строка: площади занимаемой территории (после названия области вывести отсортированный список городов каждой области);

b)  2 строка: плотности населения в них;

c)  3 строка: количества больниц, приходящихся на 1000 человек;

d)  4 строка: количества ВУЗов, приходящихся на 1000 человек.

Код програми:

# Ihor Mostovyi

# 29.04.2020

import numpy as np

import os, sys

def main():

input\_file = open(os.path.join(sys.path[0], "text4.txt"), 'r')

Cities = dict()

Information = list()

for line in input\_file:

if (line == "\n"):

continue

if (line.\_\_contains\_\_("Місто")):

inf = line.split()

for i in range(0, len(inf), 1):

Information.append(inf[i])

continue

strs = line.split()

key = strs[0]

value = [i for i in strs[1:len(strs)]]

Cities[key] = value

input\_file.close()

print(str(Cities))

Set\_areas = set()

for key, value in Cities.items():

Set\_areas.add(value[0])

Area\_square = dict()

Area\_density = dict()

Area\_hospital = dict()

Area\_university = dict()

for area in Set\_areas:

Area\_square[area] = list()

Area\_density[area] = list()

Area\_hospital[area] = list()

Area\_university[area] = list()

for key, value in Cities.items():

List = Area\_square[value[0]]

List.append([key, float(value[2])])

Area\_square[value[0]] = List

List = Area\_density[value[0]]

List.append([key, float(value[1]) / float(value[2])])

Area\_density[value[0]] = List

List = Area\_hospital[value[0]]

List.append([key, (1000 \* float(value[3])) / float(value[1])])

Area\_hospital[value[0]] = List

List = Area\_university[value[0]]

List.append([key, (1000 \* float(value[3])) / float(value[1])])

Area\_university[value[0]] = List

outputFile = open(os.path.join(sys.path[0], 'task3\_result.txt'), 'w')

print("\nAreas\_square:")

outputFile.write("\nAreas\_square:\n")

printing(Area\_square, outputFile)

print("\nAreas\_density:")

outputFile.write("\nAreas\_density:\n")

printing(Area\_density, outputFile)

print("\nArea\_hospital:")

outputFile.write("\nArea\_hospital:\n")

printing(Area\_hospital, outputFile)

print("\nArea\_university:")

outputFile.write("\nArea\_university:")

printing(Area\_university, outputFile)

outputFile.close()

return None

def printing(Areas, output\_file):

for key, value in Areas.items():

print(str(key) + ":")

output\_file.write(str(key) + ":")

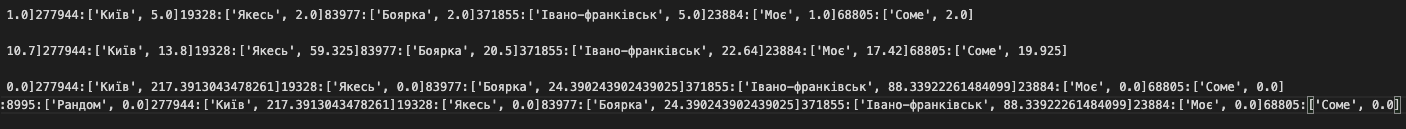
for city in value:

print(str(city))

output\_file.write(str(city))

return None

main()

Результат роботи:

***Висновки:*** Було вивчено теоретичні відомості з теми «Одномірні і багатовимірні масиви в Python. Знайомство з NumPy: створення масивів, застосування базових операцій і функцій. Робота з файлами». Було набуто практичні навички створення програм з використанням розширень мови Python.