НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №6
із дисципліни «Програмування»
на тему
«**РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ**»

Виконав:
студент групи КМ-83
Касіч Б.В.

Керівник:
ст.вик.
Дрозденко О.М.

Київ — 2018

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Мета роботи

Вивчення роботи з одновимірними і двовимірними масивами із застосуванням рекурсивних функцій.

1.2 Що потрібно знати

- методи створення масивів;
- модулі та їх функції / методи (*array* і *NumPy*);
- базові операції з масивами;
- рекурсивні функції для роботи з масивами.

1.3 Узагальнене формулювання завдання до лабораторної роботи

1) Вивчити принципи і способи створення одновимірних і двовимірних масивів в мові Python.
2) Розробити програми відповідно до варіанта завдання.
3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.
4) Показати розроблену програму викладачеві.
5) Письмово відповісти на Питання для самоперевірки.
6) Оформити звіт відповідно до вимог.

Завдання на виконання лабораторної роботи складається з 2-х частин:
− робота з одновимірними масивами (векторами); функції для обробки векторів повинні бути рекурсивними;
− робота з матрицями.

Перше завдання: Дан одновимірний масив числових значень, що нараховує n елементів. Суму елементів масиву і кількість додатних елементів поставити на перше і друге місце.

Друге завдання: Виконати обробку елементів прямокутної матриці A, що має n рядків і m стовпців. Знайти суму елементів всієї матриці. Визначити, яку частку в цій сумі становить сума елементів кожного стовпця. Результат оформити у вигляді матриці з n+1 рядків і m стовпців.

2. ОПИС ПРОГРАМ

2.1 Перша програма

За допомогою команди print ми вітаємо користувача та надаємо йому інформацію про дану програму. Імпортуємо з модуля re метод match задля подальшої валідації даних. Імпортуємо бібліотеку numpy. Визначаємо функцію get_elems завдяки якій користувач може вводити свої числа які методом append додаються у порожній список. Валідація виконана за допомогою умовних конструкцій if-else та визначених раніше шаблонів. Визначаємо функцію get_all_positives , яка поверне нам кількість додатніх чисел у масиві. Виконано за допомогою ітератора for та зростаючої змінної counter. Далі визначаємо порожній список, шаблони валідаціїї даних, викликаємо функцію get_elems, присвоюємо змінній sum_elems значення суми елементів, присвоюємо змінній all_positives відповідну функцію. Далі за допомогою команди print виводимо масивом суму елементів, кількість додатніх елементів та сам масив. Потім за допомогою методу insert вставляємо наші дані на 0 та 1 позиції відповідно і виводимо результуючий масив.

Далі за допомогою команди input ми запитуємо у користувача , хоче він завершити програму чи ні. Якщо хоче , то він має написати stop і програма завершиться не почавши цикл while. В цьому циклі безпосередньо знаходиться увесь код без визначення функцій.

2.2 Друга програма

За допомогою команди print ми вітаємо користувача та надаємо йому інформацію про дану програму. Імпортуємо з модуля re метод match задля подальшої валідації даних. Визначаємо шаблони валідації. Визначаємо функції columns та rows , які за допомогою циклічної валідації(while) повернуть значення кількости стовпців та рядків у матриці відповідно. Визначаємо функцію elems, яка за допомогою подвійного ітерування(for) та циклічної валідації дозволять користувачу ввести кожний елемент матриці и потім поверне саму матрицю. Присвоюємо змінним значення перших двух функціх , створюємо матрицю n x m, яка складається з нулів , а потім викликаємо функцію elems , яка безпосередньо допоможе змінити нулі на інші числа. Далі за допомогою ітератора for знаходимо суму всіх елементів та додаємо до відповідного вкладеного масиву за допомогою методу append і. Якщо сума елементів не дорівнює нулю , то ми виводимо її значення , а в іншому випадку говоримо про те , що частку знайти неможливо. Далі за допомогою ітератора for та методу try-except присвоюємо останньому елементу кожного вкладеного масива значення суми відповідного стовпця.
Далі за допомогою подвійного ітерування виводимо поелементно нашу матрицю с додатковим рядком сум елементів стовпців.
Далі за допомогою команди input ми запитуємо у користувача , хоче він завершити програму чи ні. Якщо хоче , то він має написати stop і програма завершиться не почавши цикл while, в якому знаходиться увесь код без функцій та шаблонів.

3. РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

3.1 Усі можливі результати першої програми продемонстровано на Рисунку 3.1

```
        ======================================================================
                              ЛАБОРАТНОРНА РОБОТА №6
                            РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ
                          Завдання на одновимірні масиви
                        АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
        ======================================================================
Добрий день! Дана програма дозволить вам ввести поелементно массив дійсних чисел,
а потім програма виведе суму всіх елементів та кількість додатніх, поставивши їх
на 0 и 1 позицію у массиві відповідно

Type 'eno' if u want to stop typing. Type your float number: пцу
Type digits only
Type 'eno' if u want to stop typing. Type your float number: 3
Type 'eno' if u want to stop typing. Type your float number: -2.5
Type 'eno' if u want to stop typing. Type your float number: 4
Type 'eno' if u want to stop typing. Type your float number: -2
Type 'eno' if u want to stop typing. Type your float number: eno
sum: [2.5]
positives: [2]
your array: [ 3.  -2.5  4.  -2. ]
resulting array: [ 2.5  2.   3.  -2.5  4.  -2. ]
Type 'stop' if u want to stop the prog: stop
>>>
```

Рисунок 3.1

3.2 Усі можливі результати другої програми продемонстровано на Рисунку 3.2

```
        ======================================================================
                              ЛАБОРАТНОРНА РОБОТА №6
                            РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ
                      Завдання на двовимірні масиви (матриці)
                      АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
        ======================================================================
Добрий день! Дана програма дозволить вам створити матрицю дійсних чисел поелементно.
А потім вона виведе суму усіх елементів матриці, саму матрицю з додатковим рядком, в якому
знаходятся числа які визначають частку суми елементів стовця в основній сумі елементів матриці.

Number of columns: 3
Number of rows: 2
Print your float number to a[0][0]: 1
Print your float number to a[1][0]: 2
Print your float number to a[0][1]: 3
Print your float number to a[1][1]: 3
Print your float number to a[0][2]: 2
Print your float number to a[1][2]: 1

Sum of all elements=12.0

1.0 3.0 2.0
2.0 3.0 1.0
0.25 0.5 0.25
print 'stop' if u want to stop the prog:  stop
>>>
```

Рисунок 3.2

Текст першої програми:

```
print("""
        ======================================================================
                                ЛАБОРАТНОРНА РОБОТА №6
                              РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ
                            Завдання на одновимірні масиви
                         АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
        ======================================================================
Добрий день! Дана програма дозволить вам ввести поелементно массив дійсних чисел,
а потім програма виведе суму всіх елементів та кількість додатніх, поставивши їх
на 0 и 1 позицію у массиві відповідно
                                                                                """)
from re import match
import numpy as np
def get_elems():
    user_input=input("Type 'eno' if u want to stop typing. Type your float number: ")
    if user_input=="eno":
        return
    else:
        while not match(pattern_float,user_input) and not match(pattern_int,user_input):
            print("Type digits only")
            return get_elems()
    a.append(float(user_input))
    return get_elems()

def get_all_positives(a):
    counter=0
    for i in a:
        if i > 0:
            counter+=1
    return counter



a=[]
pattern_int = r"^[-\d]\d*$"
pattern_float = r"^[-\d]\d*\.\d*$"
get_elems()
sum_elems = sum(a)
all_positives=get_all_positives(a)
print("sum:",np.array([sum_elems]))
print("positives:",np.array([all_positives]))
print("your array:",np.array(a))
a.insert(0,all_positives)
a.insert(0,sum_elems)
print("resulting array:",np.array(a))
```

```
answer = input("Type 'stop' if u want to stop the prog: ")
while answer != 'stop':
    a = []
    get_elems()
    sum_elems = sum(a)
    all_positives = get_all_positives(a)
    print("sum:", np.array([sum_elems]))
    print("positives:", np.array([all_positives]))
    print("your array:", np.array(a))
    a.insert(0, all_positives)
    a.insert(0, sum_elems)
    print("resulting array:", np.array(a))
    answer = input("Type 'stop' if u want to stop the prog: ")
```

Текст другої програми:

```
print("""
        ======================================================================
                                ЛАБОРАТНОРНА РОБОТА №6
                              РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ
                        Завдання на двовимірні масиви (матриці)
                        АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
        ======================================================================
Добрий день! Дана програма дозволить вам створити матрицю дійсних чисел поелементно.
А потім вона виведе суму усіх елементів матриці, саму матрицю з додатковим рядком, в якому
знаходятся числа які визначають частку суми елементів стовця в основній сумі елементів матриці.
                                                                                """)
from re import match
pattern_int = r"^[-\d]\d*$"
pattern_float = r"^[-\d]\d*\.\d*$"

def columns():
    m = input("Number of columns: ")
    while not match(pattern_int, m):
        m = input("Type integer only please:  ")
    m = int(m)
    while m <= 0:
        m = input("Type integer higher than zero please:  ")
        while not match(pattern_int, m):
            m = input("Type integer only please: ")
        m=int(m)
    return int(m)

def rows():
    n = input("Number of rows: ")
    while not match(pattern_int, n):
        n = input("Type integer only please:  ")
    n = int(n)
    while n <= 0:
        n = input("Type integer higher than zero please:  ")
        while not match(pattern_int, n):
            n = input("Type integer only please: ")
        n=int(n)
    return int(n)
```

```
def elems(a):
    for i in range(len(a)):
        for j in range(len(a[i])):
            message = "Print your float number to a[" + str(j) + "][" + str(i) + "]: "
            a[i][j] = input(message)
            while not match(pattern_int,a[i][j]) and not match(pattern_float,a[i][j]):
                print("Print correct data")
                a[i][j]=input(message)
            a[i][j] = float(a[i][j])
    return a

m=columns()
n=rows()
a = [[0] * n for i in range(m)]
elems(a)
s=0
for i in range(len(a)):
    temp=sum(a[i])
    s+=temp
    a[i].append(temp)
if s!=0:
    print()
    print("Sum of all elements="+str(s))
    print()
else:
    print()
    print("Sum of all elements equal to zero \nso we cannot find the quotient values in the columns")
for i in range(len(a)):
    try:
        a[i][-1]=round(a[i][-1]/s,3)
    except ZeroDivisionError:
        print()
for j in range(n + 1):
    for i in range(m):
        print(a[i][j], end=" ")
    print()
```

```
answer=input("print 'stop' if u want to stop the prog:  ")
while answer!="stop":
    m = columns()
    n = rows()
    a = [[0] * n for i in range(m)]
    elems(a)
    s = 0
    for i in range(len(a)):
        temp = sum(a[i])
        s += temp
        a[i].append(temp)
    if s != 0:
        print()
        print("Sum of all elements=" + str(s))
        print()
    else:
        print()
        print("Sum of all elements equal to zero \nso we cannot find the quotient values in the columns")
    for i in range(len(a)):
        try:
            a[i][-1] = round(a[i][-1] / s, 3)
        except ZeroDivisionError:
            print()
    for j in range(n + 1):
        for i in range(m):
            print(a[i][j], end=" ")
        print()
    answer = input("print 'stop' if u want to stop the prog:  ")
```

Відповіді до контрольних запитань

1. Як створюється одновимірний масив? Як отримати доступ до будь-якого елементу масиву?
Створюється и індксується так само як список
2. Як створюється матриця? Як отримати доступ до будь-якого елементу матриці?
Матриця створюється генератором двувимірних масивів. Доступ до елемента в мариці M: M[row][column], де нумерація рядків і стовпчиків починається з нуля
3. Які існують модулі для роботи з масивами?
Модуль array містить визначення типу послідовності array.array, здатної зберігати числа або символи досить економним способом. Масиви використовуються, коли потрібно досягти високої швидкості роботи. В інших випадках масиви можна замінити іншими типами даних: списками, кортежами, рядками.
4. Для чого можна використовувати генератори списків в масивах?
Щоб зробити масив за певним шаблоном , або зробити його двовимірним
5. Методи додавання елементів в масив.
array.append(x) −додавання елемента в кінець масиву.
array.extend(iter) − додавання елементів із об'єкта в масив.
array.fromlist(список)−додавання елементів зі списку.
array.insert(n,x) −включити новий пункт зі значенням x в масиві перед номером n.
6. Що таке осі масиву?
Вказавши параметр axis, можна застосувати операцію для зазначеної осі масиву. Коли axis=0– звернення іде до стовпчика матриці, а коли axis=1– до рядків
7. Як можна створити послідовність чисел в масиві?
Згенерувати будь-яким генератором, або ввести вручну.
8. Яка функція створює одиничну матрицю?
Функція eye()
9. Як можна змінити форму матриці?
Додати або видалити рядок чи стовпчик
10. Як можна виконати об'єднання і розбиття матриці?
Приклад множення матриць
[[M[row][col]*N[row][col] for col in range(len(col))] for row in range(len(row))]
11. Як можна створити масив з випадкових елементів?
Найпростіший спосіб задати масив з випадковими елементами – використовувати функцію sample. Чи використовувати модуль random.

ЗМІСТ