

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра програмної інженерії та інформаційних технологій управління

Звіт з лабораторної роботи №5

З дисципліни “Основи програмування Python”

Виконала:

студентка групи КН-221д

Хорєшко К.О.

Перевірила:

Доцент: Козуля М.М.

Харків 2022

**Мета:** Навчитися використовувати функції користувача та порівнювати продуктивність коду.

Для вибору номера варіанта я скористалася формулою:  $N=(n+4)\%5+1$ , де  $N$  – номер варіанту,  $n$  – номер в алфавіті першої літери вашого імені (наприклад, для імені «Ксенія» це буде  $N = (11 + 4) \% 5 + 1 = 1$ ).

### Завдання на 65D:

1. Для задачі свого варіанту зробити 3-4 функції, використовуючи різні підходи (наприклад, різні види циклів, спискові включення, зрізи, використанні вбудованих функцій).
2. Протестувати отримані функції на невеликих списках. Для тестування використовуйте оператори `assert`. Ці оператори зупиняють програму та видають помилку `AssertionError`, якщо їхня умова `False`.
3. Згенерувати список розміром 10000-100000 елементів та протестувати час виконання. Зробити висновок про швидкість використаних підходів.

### Варіант 1

Створити послідовність із  $n$  послідовних елементів, що діляться на 3 та 5.  $n$  вводиться з клавіатури.

```
from random import randint

def f1(n):

    x = 1

    list1 = []

    for i in range (n):

        status = True

        while status:

            if x % 3 == 0 and x % 5 == 0:
```

```
        list1.append(x)

        status = False

        x += 1

    else:

        x += 1

    return list1

def f2(n):

    list_of_numbers = [i for i in range (n*100)]

    list2 = [x for x in list_of_numbers if (x % 3 == 0 and x % 5 == 0
and x )]

    list2 = list2[0:n]

    return list2

def f3(n):

    list3 = []

    x = 0

    i = 0

    while i < n:

        x += 3*5

        list3.append(x)

        i += 1

    return list3

def f4(n):

    random_list = []

    i = 0
```

```

while i < n:

    x = randint(15, n*15)

    if x not in random_list and x % 15 == 0 and x // 15 == i+1:

        random_list.append(x)

        i += 1

return random_list

n = int(input("Enter n: "))

print(f1(n))

print(f2(n))

print(f3(n))

print(f4(n))

print(" ")

```

Рис. 1 – Код для знаходження послідовності із n послідовних елементів, що діляться на 3 та 5(представлення п.1)

```

Enter n: 5
[15, 30, 45, 60, 75]
[15, 30, 45, 60, 75]
[15, 30, 45, 60, 75]
[15, 30, 45, 60, 75]

```

Рис. 2 – Виконання програми(п.1)

```

assert f1(n) == f2(n) == f3(n) == f4(n)

print('Looks like you pass all tests :)\n')

```

Рис. 3 – Код для тестування результату, чи однаковий результат виводять функції(представлення п.2)

```
Enter n: 7
[15, 30, 45, 60, 75, 90, 105]
[15, 30, 45, 60, 75, 90, 105]
[15, 30, 45, 60, 75, 90, 105]
[15, 30, 45, 60, 75, 90, 105]

Looks like you pass all tests :)
```

Рис. 4 – Виконання програми(п.2)

```
def f1(n):

    x = 1

    list1 = []

    for i in range (10000,n):

        status = True

        while status:

            if x % 3 == 0 and x % 5 == 0:

                list1.append(x)

                status = False

                x += 1

            else:

                x += 1

        return list1

def f2(n):

    list_of_numbers = [i for i in range (10000,n*100)]

    list2 = [x for x in list_of_numbers if (x % 3 == 0 and x % 5 == 0
and x )]

    list2 = list2[0:n]

    return list2
```

```
def f3(n):

    list3 = []

    x = 0

    i = 10000

    while i < n:

        x += 3*5

        list3.append(x)

        i += 1

    return list3


def f4(n):

    random_list = []

    i = 1000

    while i < n:

        x = randint(15, n*15)

        if x not in random_list and x % 15 == 0 and x // 15 == i+1:

            random_list.append(x)

            i += 1

    return random_list


print('Let\'s check time' )

import time

start = time.time()

f1(randint(10000,100000))

stop = time.time()
```

```

print(f"Time for first function is {round(stop - start,2)} sec")

start = time.time()

f2(randint(10000,100000))

stop = time.time()

print(f"Time for second function is {round(stop - start,2)} sec")

start = time.time()

f3(randint(10000,100000))

stop = time.time()

print(f"Time for third function is {round(stop - start,2)} sec")

start = time.time()

f4(1200)

stop = time.time()

print(f"Time for forth function is {round(stop - start,2)} sec")

```

Рис. 3 – Код для тестування часу виконання функцій(представлення п.3)

```

Let's check time
Time for first function is 0.54 sec
Time for second function is 0.78 sec
Time for third function is 0.05 sec
Time for forth function is 28.94 sec

```

Рис. 4 – Виконання програми(п.3)

### Висновок стосовно продуктивності наведених функцій:

Четвертій функції було надано значення набагато менше, порівняно з іншими функціями з тої причини, що незважаючи на те, що вона правильно працює, виконання коду займає забагато часу. По моїм підрахункам, на основі результату виконання коду на наведеному вище проміжку, час виконання цієї програми на проміжку до 10000 елементів  $\approx 40$  хв., а на проміжку до 100000 елементів  $\approx 3$  год., тому цей спосіб є найповільнішим. Найшвидшою є третя функція(0.05с).

## Завдання на 75-100:

1. Завдання на 65D
2. Розв'язати одну (max 89 балів) або дві (max 100 балів) задачі на списки (list) з колекції codewars - <https://www.codewars.com/collections/python-fundamentals-21>
3. Порівняти продуктивність свого рішення та декількох рішень інших користувачів

### Twisted Sum

#### DESCRIPTION:

Find the sum of the digits of all the numbers from 1 to N (both ends included).

#### Examples

# N = 4

$$1 + 2 + 3 + 4 = 10$$

# N = 10

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + (1 + 0) = 46$$

# N = 12

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + (1 + 0) + (1 + 1) + (1 + 2) = 51$$

```
def f1(n):  
  
    list1 = []  
  
    for i in range (1,n+1):  
  
        if i // 10 == 0:
```



```

        list1.append(i)

    else:

        while i // 10 > 0:

            a = i % 10

            list1.append(a)

            i = i // 10

        list1.append(i)

    print(' + '.join(map(str, list1)), '=', sum(list1))

n = int(input("Enter n: "))

f1(n)

```

Рис. 5 – Код розрахунку суми(Twisted Sum, п.2)

```

Enter n: 12
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 0 + 1 + 1 + 1 + 2 + 1 = 51

```

Рис. 6 – Виконання програми(Twisted Sum,п.2)

```

def f1(n):

    list1 = []

    for i in range (1,n+1):

        if i // 10 == 0:

            list1.append(i)

        else:

            while i // 10 > 0:

                a = i % 10

                list1.append(a)

                i = i // 10

            list1.append(i)

```

```

        #print(' + '.join(map(str, list1)), '=', sum(list1))

def f2(n):

    #n = input("Please input an integer : ")

    n = int(n)

    item = range(1, n + 1)

    for i in item:

        sum = 0

        for j in range(1, i + 1):

            sum = sum + j

        #print(str(sum))

def f3(n):

    #x = input("Please input an integer: ")

    n = int(n)

    for i in range(1, n+1):

        nums = range(1, i+1)

        sum(nums)

        #print(' + '.join(map(str, nums)), '=', sum(nums))

n = 0

assert f1(n) == f2(n) == f3(n)

print('Looks like you pass all tests :)\n')

print('Let\'s check time' )

import time

```

```

import random

start = time.time()

f1(random.randint(1,10000))

stop = time.time()

print(f"Time for first function is {round(stop - start,2)} sec")

start = time.time()

f2(random.randint(1,10000))

stop = time.time()

print(f"Time for second function is {round(stop - start,2)} sec")

start = time.time()

f3(random.randint(1,10000))

stop = time.time()

print(f"Time for third function is {round(stop - start,2)} sec")

```

Рис. 7 – Код перевірки продуктивності моєї функції та представлень інших користувачів(Twisted Sum, п.3)

```

Looks like you pass all tests :)

Let's check time
Time for first function is 0.01 sec
Time for second function is 6.65 sec
Time for third function is 0.42 sec

```

Рис. 8 – Виконання програми(Twisted Sum,п.3)

### **Висновок стосовно продуктивності наведених функцій(Twisted Sum):**

Я вважаю, що моє представлення рішення цього завдання є кращим з тієї причини, що я виконувала завдання саме за допомогою списків. До того ж моя функція виконується найшвидше (0.01с). Найповільнішою є друга функція(6.65с).

## Sum two arrays

### DESCRIPTION:

Your task is to create a function called `addArrays`, which takes two arrays consisting of integers, and returns the sum of those two arrays.

The twist is that (for example) `[3,2,9]` does not equal  $3 + 2 + 9$ , it would equal `'3' + '2' + '9'` converted to an integer for this kata, meaning it would equal 329. The output should be an array of the sum in a similar fashion to the input (for example, if the sum is 341, you would return `[3,4,1]`). Examples are given below of what two arrays should return.

`[3,2,9],[1,2] --> [4,4,9]`

`[4,7,3],[1,2,3] --> [5,9,6]`

`[1],[5,7,6] --> [6,7,6]`

```
def function(n1,n2):  
  
    list1 = []  
  
    for i in range (n1):  
  
        num1 = int(input("Enter the element of first list: "))  
  
        list1.append(num1)  
  
  
    list2 = []  
  
    for i in range (n2):  
  
        num2 = int(input("Enter the element of second list: "))  
  
        list2.append(num2)  
  
  
    print(" ")
```

```
print("First list: " + str(list1))

print("Second list: " + str(list2))


if n1 > n2:

    n = n1

    for i in range (n2,n):

        list2.append(0)

else:

    n = n2

    for i in range (n1,n):

        list1.append(0)


list_of_sum = []

for j in range (n):

    num = list1[j]+list2[j]

    list_of_sum.append(num)

print(" ")

print("List of sum: " + str(list_of_sum))


n1 = int(input("Enter range of list: "))

n2 = int(input("Enter range of list: "))

function(n1,n2)
```

Рис. 9 – Код розрахунку суми(Sum two arrays, п.2)

```
Enter range of list: 3
Enter range of list: 1
Enter the element of first list: 5
Enter the element of first list: -2
Enter the element of first list: 1
Enter the element of second list: 4

First list: [5, -2, 1]
Second list: [4]

List of sum: [9, -2, 1]
```

Рис. 10 – Виконання програми(Sum two arrays, п.2)

```
def function(n1,n2):

    list1 = []

    for i in range (n1):

        num1 = random.randint(-10,10)

        list1.append(num1)

    list2 = []

    for i in range (n2):

        num2 = random.randint(-10,10)

        list2.append(num2)

    #print(" ")

    #print("First list: " + str(list1))

    #print("Second list: " + str(list2))

    if n1 > n2:

        n = n1

        for i in range (n2,n):
```

```

        list2.append(0)

    else:

        n = n2

        for i in range (n1,n):

            list1.append(0)


list_of_sum = []

for j in range (n):

    num = list1[j]+list2[j]

    list_of_sum.append(num)

#print(" ")

#print("List of sum: " + str(list_of_sum))

def f2(n1,n2):

    #n1=int(input())

    num1 = list(map(int, input(random.randint(-10,10)).split()))

    #n2=int(input())

    num2 = list(map(int, input(random.randint(-10,10)).split()))

    sum=[]

    for i in range(0,n1):

        sum.append(num1[i]+num2[i])

    #for element in sum:

        #print(element, end=" ")

```

```

#print("")

print('Let\'s check time' )

import time

import random

start = time.time()

function(random.randint(1,1000),random.randint(1,1000))

stop = time.time()

print(f"Time for first function is {round(stop - start,2)} sec")

start = time.time()

f2(random.randint(1,1000),random.randint(1,1000))

stop = time.time()

print(f"Time for first function is {round(stop - start,2)} sec")

```

Рис. 11 – Код розрахунку суми(Sum two arrays, п.3)

```

Let's check time
Time for first function is 0.01 sec
-7

```

Рис. 12 – Виконання програми(Sum two arrays, п.3)

### **Висновок стосовно продуктивності наведених функцій(Twisted Sum):**

Я порівняла свою функцію лише з однією функцією іншого користувача, бо всі рішення були з використанням проініціалізованих масивів вже у програмному коді, що я вважаю незручним з точки зору користувача, а порівнювати таке рішення з моїм - некоректно. Вважаю, що моє представлення рішення цього завдання є кращим, ніж другий спосіб, з тієї причини, що моя функція може працювати з списками різної довжини, тому я не використовувала тестування оператором assert. До того ж моя функція виконується за 0.01с, а друга функція видає некоректне значення



часу, що свідчить саме про помилку всередині коду(скоріш за все спричинену саме різною рандомною довжиною списків).

### **Висновок:**

Під час виконання лабораторної роботи, я визначила свій варіант роботи, використала функції користувача та порівнювала продуктивність коду.