Міністерство освіти та науки України

Національний технічний університет України

«Київський Політехнічний Інститут ім. Ігоря Сікорського»

ФТІ

Кафедра ФТЗЗІ

**Лабораторна робота №5**

з дисципліни: «Програмування 4»

на тему:

**«Використання функцій у мові Python»**

**Варіант 21**

Виконав:

Ст. гр. ФЕ-81

Пелих Валентин

Перевірив:

доцент Прогонов Д.О.

Київ 2020

Мета роботи: Оволодіння практичними навичками у використанні функцій.

**1. Порядок виконання роботи**

1. Проаналізувати умову задачі.

2. Розробити алгоритм та створити програму розв’язання задачі згідно з номером варіанту.

3. Результати роботи оформити протоколом.

Завдання**(5А)**:

Дано натуральне число , цілі числа . Розглянути відрізки послідовності  (послідовності членів, що йдуть поруч), що складаються з повних квадратів. Отримати найбільшу з довжин цих відрізків. (Визначити функцію, що дозволяє розпізнавати повні квадрати.)

Код реалізації **(5A)**:

class Exception(Exception):

def init(self, text):

self.txt = text

def check\_perfect\_square(a):

if (a == 1):

print("Processing Error!")

raise Exception("The 'a' must not equal to 1!")

x = a // 2

seen = set([x])

while x \* x != a:

x = (x + (a // x)) // 2

if x in seen:

return False

seen.add(x)

return True

try:

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

an = []

print("Enter non-negative numbers to sequence.")

for i in range(0, N):

i = int(input("Elem is:" ))

an.append(i)

print("Your sequence is: ")

print(an, "\n")

print("Checking if some elements here is perfect square...")

for i in range(0, N):

print(an[i], "-", check\_perfect\_square(an[i]), end="; ")

print("\n")

segments = []; max\_seg = []

flow = False; a = 0

for i in range(0, N):

check = check\_perfect\_square(an[i])

if (check == True):

segments.append(an[i])

flow = True

else:

segments.append(0)

flow = False

if (flow == True):

a = a + 1

elif (flow == False):

max\_seg.append(a)

a = 0

max\_seg.append(a)

print("Max Segments are:", max\_seg, "\n")

max\_s = max(max\_seg)

print("The largest segment with perfect squares is: ", max\_s)

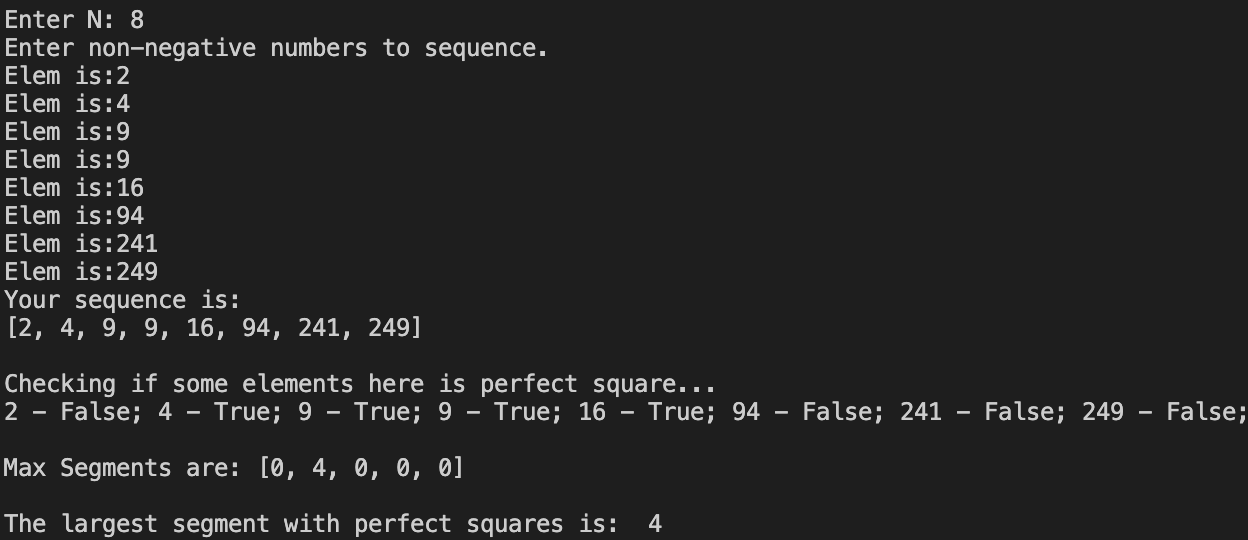
except Exception as non1:

print(non1)

except ValueError:

print("Entered elements must be non-negative!")

Приклад виконання програми**(5A)**:



Завдання**(5B)**:

5.Дано натуральні числа . Отримати , де 

 — залишок від ділення  на 10. Використати програму, що містить в собі рекурсивну функцію обчислення .

Код реалізації **(5B)**:

print("Enter a, c, m. They must be natural numbers (positive integers).")

print("Otherwise - it will be converted using abs().")

a = abs(int(input("Enter a: " )))

c = abs(int(input("Enter c: " )))

m = abs(int(input("Enter m: " )))

def f(n):

if (0 <= n <= 9):

return n

else:

res = g(n) \* f(n-1-g(n)) + n

return res

def g(n):

res = (a\*n + a\*c) % 10

return res

Res = f(m)

print("Result is: ", Res)

Приклад виконання програми**(5B)**:

