***Лабораторна робота № 2***

***Тема:*** Розгалуження. Цикли. Рекурсивні функції.

***Мета:*** набути практичні навички створення програм, що використовують зазначені конструкції, їх написання та налагодження на мові програмування Python.

***Короткі теоретичні відомості***

Синтаксис оператора розгалуження **if...else**:

if умова\_1:

блок\_операторів\_1

if: умова\_2:

блок\_операторів\_2

else:

блок\_операторів\_3

Синтаксис циклу **for...in...else** (завдання значень лічильника, використовуючи функцію range).

Слід враховувати, що функція range може використовуватися з одним або декількома параметрами:

* range (max) – повертає послідовність елементів від 0 до max (не включаючи max) з кроком 1;
* range (min, max) – повертає послідовність елементів від min до max (НЕ включаючи max) з кроком 1;
* range (min, max, step) – повертає послідовність елементів від min до max з кроком step;

for лічильник\_циклу in range (параметри):

блок\_операторів

Інструкція **break** дозволяє перервати цикл.

Інструкція **continue** дозволяє пропустити виконання наступного після неї коду і продовжити прохід по наступним елементам.

Цикл **while...else** – конструкція, що дозволяє повторювати цикл, поки виконується певна умова. Синтаксис:

while умова:

блок\_операторів

else:

блок\_операторів

***Результати роботи:***

***Завдання 1*** – Дано натуральні числа n, . Визначити кількість членів послідовності:

е) Задовольняють умову

Код програми:

#Ihor Mostovyi

#12.04.2020

def count():

satisfiedDigits = []

n = int(input("Enter count of digits. N >= 3\n"))

if n < 3:

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

return []

check = lambda first, second, third: second < (first + second) / 2

first = int(input("enter digit\n"))

second = int(input("enter digit\n"))

for \_ in range(n - 2):

third = int(input("enter digit\n"))

if check(first, second, third) :

satisfiedDigits.append(second)

first = second

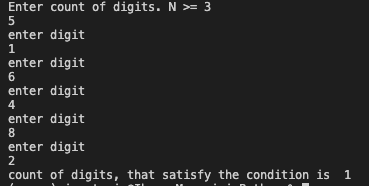
second = third

return satisfiedDigits

def main():

print("count of digits, that satisfy the condition is ", len(count()))

Результати роботи:



***Завдання 2*** – Знайти суму чисел визначених у попередній задачі.

Код програми:

#Ihor Mostovyi

#12.04.2020

from Task1 import count

def main():

sum = 0

for digit in count():

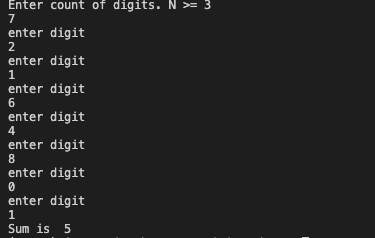
sum += digit

print("Sum is ", sum)

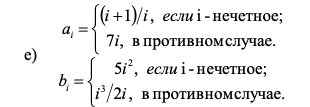
return None

main()

Результати роботи:



***Завдання 3*** – Обрахувати , де:



Код програми:

#Ihor Mostovyi

#12.02.2020

def main():

calcAi = lambda i: (i + 1) / i if i % 2 != 0 else 7 \* i

calcBi = lambda i: 5 \* (i \*\* 2) if i % 2 != 0 else (i \*\* 3) / (2 \* i)

sum = 0

for i in range(1, 31):

sum += calcAi(i) - calcBi(i)

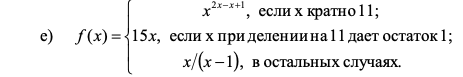
print("Sum is ", sum)

main()

Результати роботи:



***Завдання 4***- Дані натуральні числа n, *a*1,Κ,*an* . Обрахувати *f* (*a*1)+ *f* (*a*2 )+Κ+ *f* (*an* ), де:



Код програми:

def func(x):

if x % 11 == 0:

return x \*\* (x + 1)

if x % 11 == 1:

return 15 \* x

return x / (x - 1)

def main():

n = int(input("enter count of digits in sequence \n"))

if n < 0:

print("Error n < 0")

return None

sum = 0

for \_ in range(n):

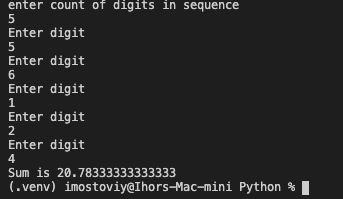
sum += func(int(input("Enter digit\n")))

print("Sum is", sum)

return None

main()

Результати роботи:



***Завдання 5*** – Дано натуральне число n і дійсні числа a1,Κ,an . Вычислить:

e)  min(a2,a4,Κ) + max(a2,a4,Κ);

Код програми:

#Ihor Mostovyi

#12.04.2020

def main():

n = int(input("enter count of digits in sequence \n"))

if n < 0:

print("Error n < 0")

return None

min = 0

max = 0

for i in range(n):

digit = int(input("Enter digit\n"))

if i == 1:

min = digit

max = digit

if i % 2 == 0:

continue

if digit < min:

min = digit

if digit > max:

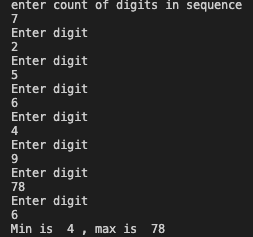
max = digit

print("Min is ", min, ", max is ", max)

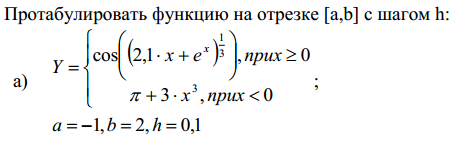
return None

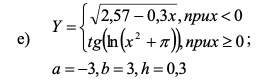
main()

Результати роботи:



***Завдання 6***:





Код програми:

from math import pi, lgamma, tan, sqrt

def function(x: float):

return sqrt(2.57 - 0.3 \* x) if x < 0 else tan(lgamma(x \*\* 2 + pi))

def main():

x: float = -3

while x <= 3:

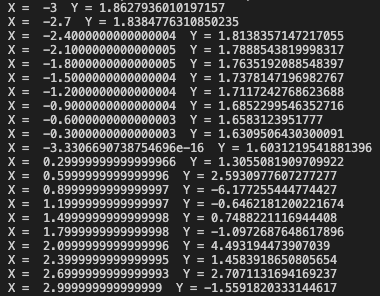
print("X = ", x, " Y =", function(x))

x += 0.3

return None

main()

Результати роботи:



***Завдання 7*** – Необхідно скласти програму розрахунку кінцевої суми та порівняння отриманого результату з контрольним значенням.



Код програми:

#Ihor Mostovyi

#12.04.2020

def sequence(i):

if i == 1:

return 1

return (2 \* i - 1) \*\* 2 + sequence(i - 1)

def controll(i):

return i \* (4 \* (i \*\* 2) - 1) / 3

def main():

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

if sequence(n) == controll(n):

print("All is ok!")

else:

print("wrong")

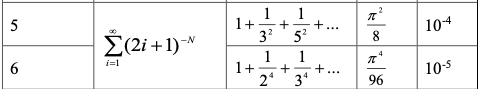
return None

main()

Результати роботи:



***Завдання 8*** – Необхідно скласти програму розрахунку нескінченної суми зворотних ступенів числового ряду. Підсумовування проводити, поки черговий член ряду по модулю не стане  менше заданої точності. Результат порівняти з точним значенням Sт, а похибку зіставити з величиною .



Код програми:

#Ihor Mostovyi

from math import pi

def main():

sum = 1

i = 2

e = 10 \*\* -4

st = (pi \*\* 2) / 8

n = 1

while abs(n) >= e:

n = (2 \* i + 1) \*\* -2

i += 1

sum += n

print("Sum is ", sum, " controll sum is ", st, ".\nE is ", e, " error is ", abs(sum - st), ".\nN is ", n, "\n")

return None

main()

Результати роботи:



***Завдання 9*** – У Змія-Горинича 2000 голів. Казковий богатир може зрубати йому одним ударом меча 33, 21, 17 або 1 голову, але при цьому у Змія виростає відповідно 48, 0, 14 або 349 голів. Якщо відрубані всі голови, то нові не виростають. Як богатирю здолати Змія?

Код програми:

# Ihor Mostovyi

# 12.04.2020

def main():

heads = 2000

hit33 = 0

hit21 = 0

hit17 = 0

hit1 = 0

while heads != 0:

if heads >= 21:

hit21 += heads // 21

heads -= (heads // 21) \* 21

continue

if heads >= 17:

hit17 += 1

heads -= 17

if heads == 0: break

heads += 14

continue

if heads >= 33:

hit33 += 1

heads -= 33

if heads == 0: break

heads += 48

continue

hit1 += 1

heads -= 1

if heads == 0: break

heads += 349

print("hit33: ", hit33, "\nhit21: ", hit21, "\nhit17: ", hit17, "\nhit1: ", hit1)

return None

main()

Результати роботи:



***Висновки:*** Було вивчено теоретичні відомості з теми «Розгалуження. Цикли. Рекурсивні функції». Було набуто практичні навички створення програм, що використовують зазначені конструкції, їх написання та налагодження на мові програмування Python.