

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Основи програмування 1.

Базові конструкції»

«Організація циклічних процесів. Ітераційні цикли»

Варіант 18

Виконав студент	ІП-11 Лесів Владислав Ігорович
Перевірив	Вітковська Ірина Іванівна

Київ 2021

Лабораторна робота 3

Організація циклічних процесів. Ітераційні цикли

Мета – вивчити особливості організації ітераційних циклів.

Варіант №18.

18. Задане дійсне число x . Послідовність a_1, a_2, \dots, a_n утворена за законом

$$a_n = x^n / (2n)! , n = 1, 2, \dots$$

Отримати суму $a_1 + a_2 + \dots + a_k$, де k - найменше ціле число, що задовольняє двом умовам: $k > 10, |a_k| < 10^{-5}$.

Постановка задачі. Результатом розв'язку є сума елементів заданої законом послідовності за виконання необхідним умов задачі, та число-умова виконання критеріїв. Для визначення результату повинне бути задане дійсне число x . Інших початкових даних для розв'язку не потрібно.

Побудова математичної моделі. Складемо таблицю імен змінних.

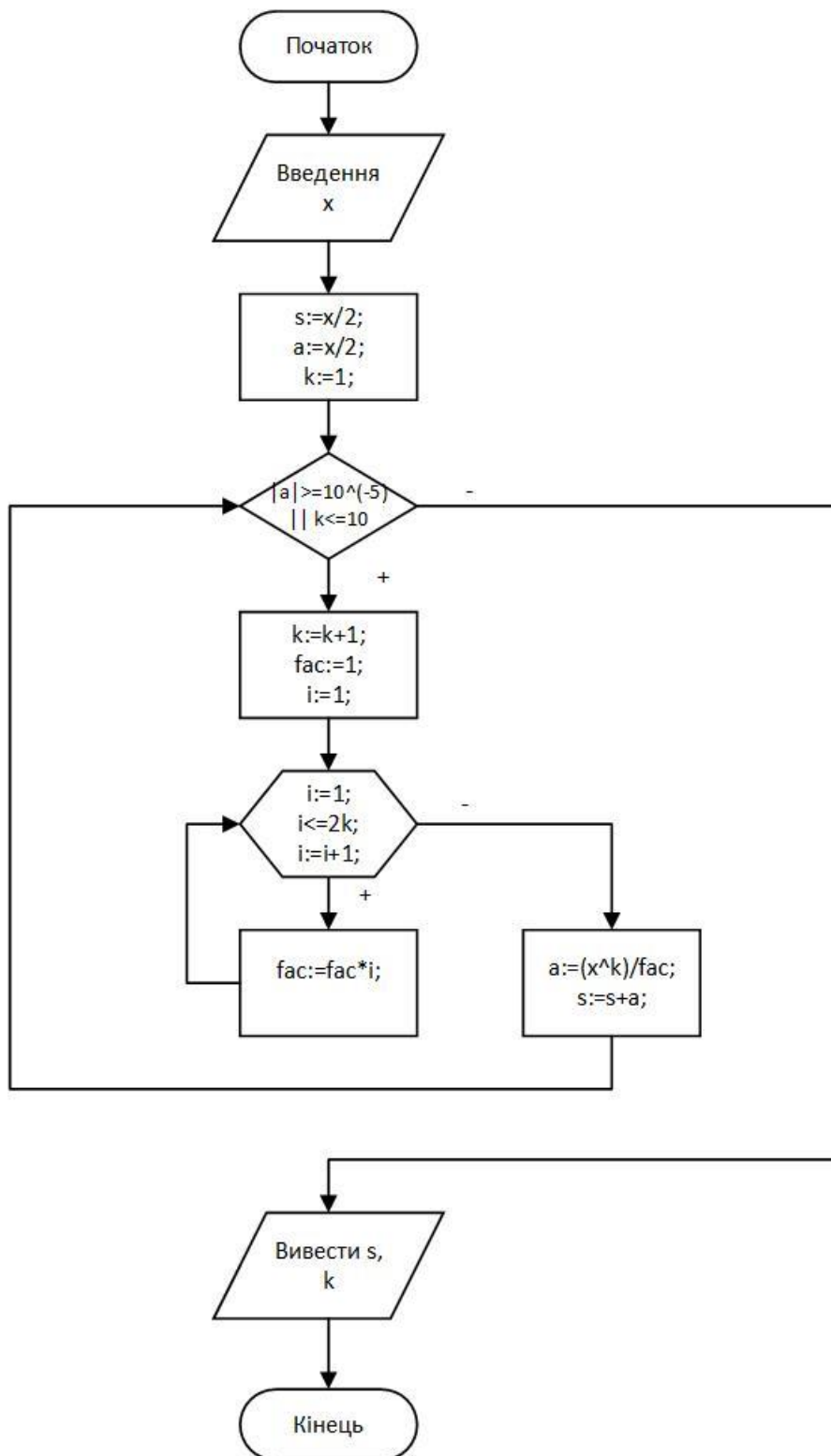
Змінна	Тип	Ім'я	Призначення
Задане дійсне число x	Дійсний	x	Початкове дане
Елемент послідовності	Дійсний	a	Проміжне значення
Значення факторіалу числа	Цілий	fac	Проміжне значення
Сума елементів послідовності	Дійсний	s	Результат
Число, яке задовільняє умови задачі	Цілий	k	Результат

Математичне формулювання задачі зводиться до перевірки виконання заданих умов. Якщо модуль значення елемента послідовності на кроці k більший або рівний за 10^{-5} , тобто $|a| < 10^{-5}$, а саме число $k > 10$, отримуємо виконання умов, що спричинить завершення перевірки. У іншому випадку одна з умов не

виконується, тому додаємо елемент до суми елементів та продовжуємо перевірку в циклі.

Програмні специфікації запишемо у графічній формі у вигляді блок-схеми.

Блок-схема

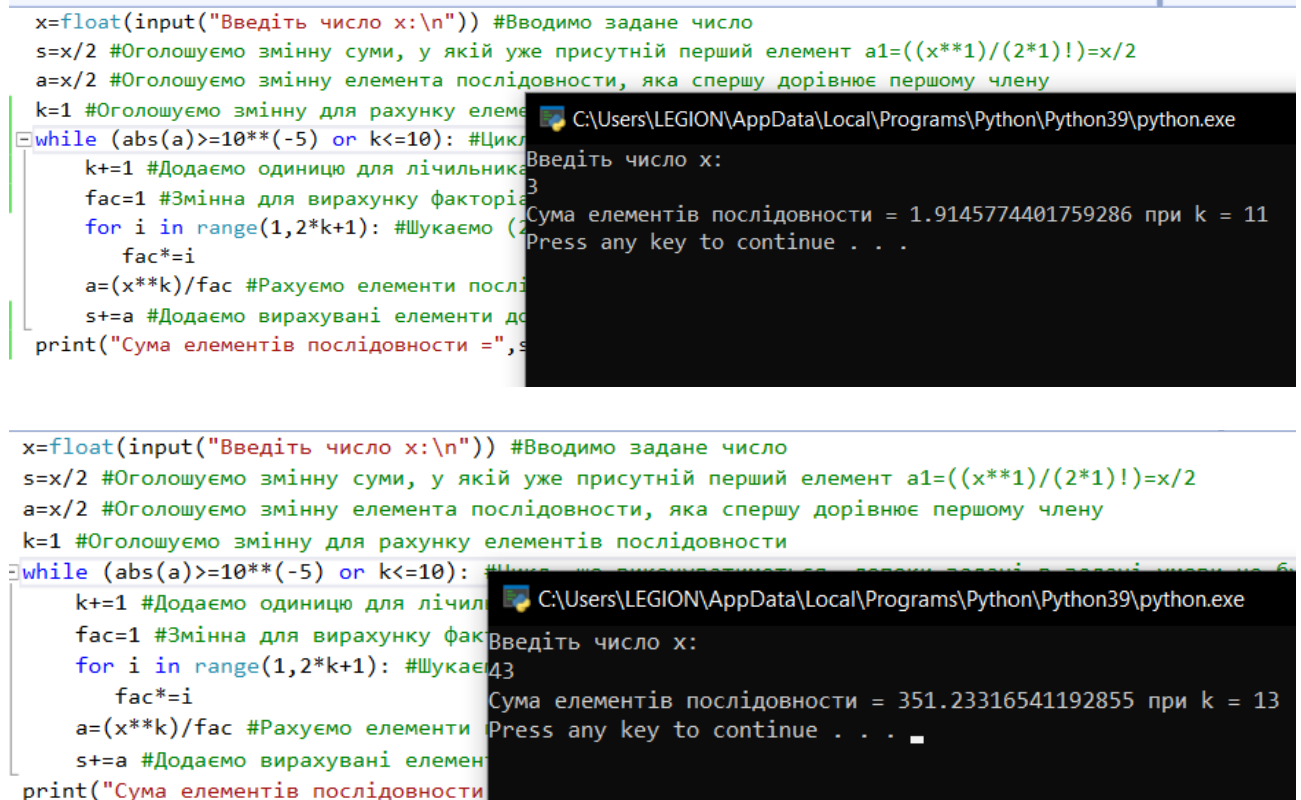


Виконання мовою Python.

Код програми:

```
x=float(input("Введіть число x:\n")) #Вводимо задане число
s=x/2 #Оголошуємо змінну суми, у якій уже присутній перший елемент
a1=((x**1)/(2*1!))=x/2
a=x/2 #Оголошуємо змінну елемента послідовности, яка спершу дорівнює першому члену
k=1 #Оголошуємо змінну для рахунку елементів послідовности
while (abs(a)>=10**(-5) or k<=10): #Цикл, що виконуватиметься, допоки задані в задачі умови не будуть досягнуті
    k+=1 #Додаємо одиницю для лічильника
    fact=1 #Змінна для вирахунку факторіалу числа
    for i in range(1,2*k+1): #Шукаємо (2k)!
        fact*=i
    a=(x**k)/fact #Рахуємо елементи послідовности
    s+=a #Додаємо вираховані елементи до суми
print("Сума елементів послідовности =",s,"при k =",k)
```

Випробування алгоритму.



```
x=float(input("Введіть число x:\n")) #Вводимо задане число
s=x/2 #Оголошуємо змінну суми, у якій уже присутній перший елемент a1=((x**1)/(2*1!))=x/2
a=x/2 #Оголошуємо змінну елемента послідовности, яка спершу дорівнює першому члену
k=1 #Оголошуємо змінну для рахунку елементів послідовности
while (abs(a)>=10**(-5) or k<=10): #Цикл, що виконуватиметься, допоки задані в задачі умови не будуть досягнуті
    k+=1 #Додаємо одиницю для лічильника
    fact=1 #Змінна для вирахунку факторіалу числа
    for i in range(1,2*k+1): #Шукаємо (2k)!
        fact*=i
    a=(x**k)/fact #Рахуємо елементи послідовности
    s+=a #Додаємо вираховані елементи до суми
print("Сума елементів послідовности =",s,"при k =",k)
```

C:\Users\LEGION\AppData\Local\Programs\Python\Python39\python.exe

Введіть число x: 1

Сума елементів послідовности = 1.9145774401759286 при k = 11

Press any key to continue . . .

```
x=float(input("Введіть число x:\n")) #Вводимо задане число
s=x/2 #Оголошуємо змінну суми, у якій уже присутній перший елемент a1=((x**1)/(2*1!))=x/2
a=x/2 #Оголошуємо змінну елемента послідовности, яка спершу дорівнює першому члену
k=1 #Оголошуємо змінну для рахунку елементів послідовности
while (abs(a)>=10**(-5) or k<=10): #Цикл, що виконуватиметься, допоки задані в задачі умови не будуть досягнуті
    k+=1 #Додаємо одиницю для лічильника
    fact=1 #Змінна для вирахунку факторіалу числа
    for i in range(1,2*k+1): #Шукаємо (2k)!
        fact*=i
    a=(x**k)/fact #Рахуємо елементи послідовности
    s+=a #Додаємо вираховані елементи до суми
print("Сума елементів послідовности =",s,"при k =",k)
```

C:\Users\LEGION\AppData\Local\Programs\Python\Python39\python.exe

Введіть число x: 43

Сума елементів послідовности = 351.23316541192855 при k = 13

Press any key to continue . . .

Висновок. Отже, у цій роботі я вивчив особливості організації ітераційних циклів. У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі, та блок-схему, яка пояснює логіку алгоритму. Використовуючи ітераційний цикл `while`, перевіряючи задані умови, отримуємо коректний результат.