ЕТАП №7

«Розробка та перевірка програми

РОЗРАХУНКОВО-ГРАФІЧНОЇ РОБОТИ»

з дисципліни: «Програмування» 1-й семестр

на тему: «Програма обчислення визначених інтегралів за квадратурними формулами (формули Сімпсона)»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-02  Хавронюк Б. А. | Олефір О.С. |

Київ-2020

**Розробка та перевірка програми**

1. **Текст розробленої програми**:

from math import fabs  
  
print("\n\n\n", "вас вітає програма обчислення визначених інтнгралів методом Cімпсона \n\n\n\n")  
description = """  
Програма призначена обчислювати визначені інтеграли за формулою Сімпсона.  
Користувач вводить підінтегральну формулу аналітичним способом, межі інтегрування, максимально допустиму похибку, та кількість розбиттів інтегралу.  
Програма обчислює площу визначеного інтегралу та повертає відповідне значення округлене до двох знаків після коми.  
Після чого користувач може ввести іншу функцію і програма почнеться знову, аж поки користувач не захоче вийти з програми. """  
requirements = """  
ВИМОГИ ДО ВВЕДЕННЯ  
Для корректного вводу функції використовуйте наступні оператори:  
"+" - оператор додавання  
"-" - оператор віднімання  
"\*" - оператор множення  
"/" - оператор ділення. Помітка: складні вирази потрібно брати в дужки: (sqrt(x) - 1)/(sqrt(x^2 - 1)+x\*\*2)  
"\*\*" - зведення в степінь, наприклад x\*\*3 - це х в кубі  
"pi" - число пі. Наприклад, cos(x+pi/2)  
log(x, e) - логарифм за основою e вираза в дужках. Наприклад, log(3\*x, 2) - логарифм з основою 2 числа 2x.  
 Другий параметр можна опустити, тоді це буде натуральний логарифм.  
sqrt(x) - квадратний корінь з вираза в дужках. Наприклад, sqrt(x\*\*7) - квадратний корінь з x в сьомій  
sin(x), cos(x), tan(x) - функції синус, косинус і тангенс від x відповідно.  
asin(x), acos(x), atan(x) - функції арксинус, арккосинус і арктангенс від x відповідно."""  
print(description)  
print(requirements)  
  
while True:  
  
 while True:  
 try:  
 a = float(input("веедіть початкову точку інтегруання"))  
  
 except ValueError:  
 print("неправильний формат даних ")  
 else:  
 break  
 while True:  
 try:  
 b = float(input("веедыть кінцеву точку інтегруання"))  
  
 except ValueError:  
 print("неправильний формат даних ")  
 else:  
 break  
  
 while True:  
 while True:  
 try:  
 n = int(input("введіть кількість відрізків розбиття"))  
 except ValueError:  
 print("неправильний формат даних ")  
 else:  
 break  
 if n % 2 != 0:  
 n = int(input("введіть парну кількість відрізків розбиття"))  
 else:  
 break  
 while True:  
 try:  
 maxerror = float(input("максимальна похибка"))  
  
 except ValueError:  
 print("неправильний формат даних ")  
 else:  
 break  
 while True:  
 try:  
 global f  
 f = input('\nВведіть функцію, дотримуючись вище наведених правил: ')  
 check = eval('lambda x: ' + f)  
 for i in range(int(a), int(b) + 1):  
 check(i)  
 break  
 except ZeroDivisionError:  
 print('Введена функція містить ділення на нуль. Введіть іншу функцію, дотримуючись наведених вище правил.')  
  
 except ValueError:  
 print('Введена функція не є дійсною.')  
 print('Перевірте правильність введення та введіть іншу функцію, дотримуючись наведених вище правил.')  
  
  
 def method\_Sympson(a, b, n):  
 h = (b-a) / n  
  
 function = eval('lambda x: ' + f)  
  
 if a > b:  
 print('неправильні межі')  
 return None  
 resalt = (function(a) + function(b))  
 for i in range(1,n):  
 m = a + i \* h  
  
 if i % 2 == 0:  
  
  
 resalt = resalt + 2\*function(m)  
  
 else:  
  
  
 resalt = resalt+4\* function(m)  
  
 resalt = resalt\*h/3  
  
  
 return resalt  
  
  
 def errorfinding(a, b, n ):  
  
  
 error= fabs(method\_Sympson( a, b, 2\*n) - method\_Sympson( a, b, n))  
  
 return error  
  
  
 while True:  
 method\_Sympson( a, b, n)  
 if errorfinding(a, b, n ) >= maxerror:  
 n = n \* 2  
  
 else:  
 break  
 print("отримана похибка",errorfinding(a, b, n ))  
 print("Результат" ,method\_Sympson( a, b, n))  
 print("\nБажаєте спробувати ще?", "Напишіть так або ні.", sep="\n")  
 while True:  
 ans = input()  
 if ans == "так" or ans == "ні":  
 break  
 else:  
 print("Недопустима відповідь. Введіть так або ні.")  
 if ans == "ні":  
 print("До зустрічі!")  
 break

**Експериментальні розрахунки**

Функція 1/x\*\*2

на проміжку 2 - 4

бажаний результат 1/4

в Рис 1.1 результат роботи програми

Функція x\*\*2+x\*\*3

на проміжку 1 - 9

бажаний результат 5648/3

в Рис 1.2 результат роботи програми

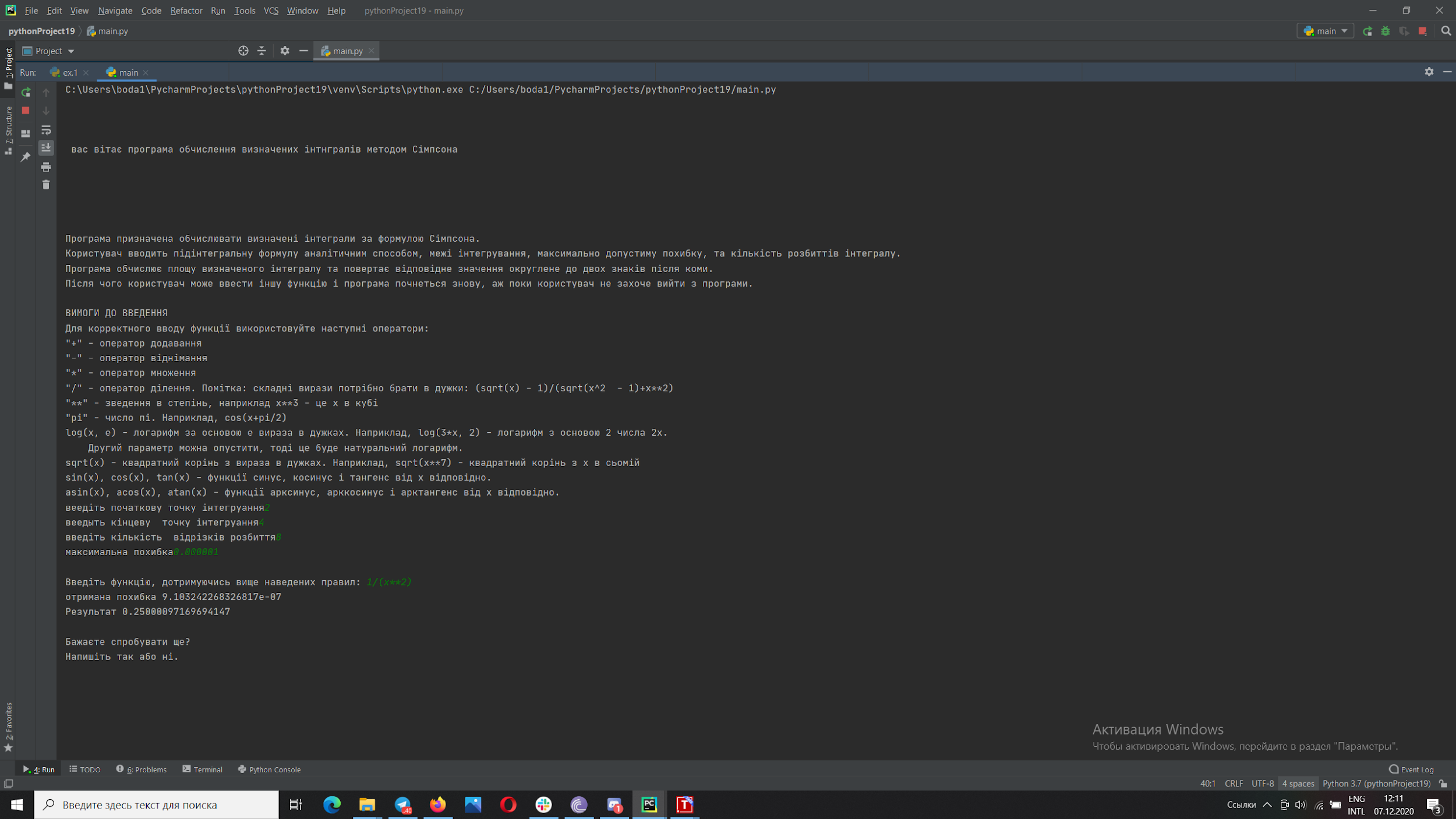


Рис 1.1

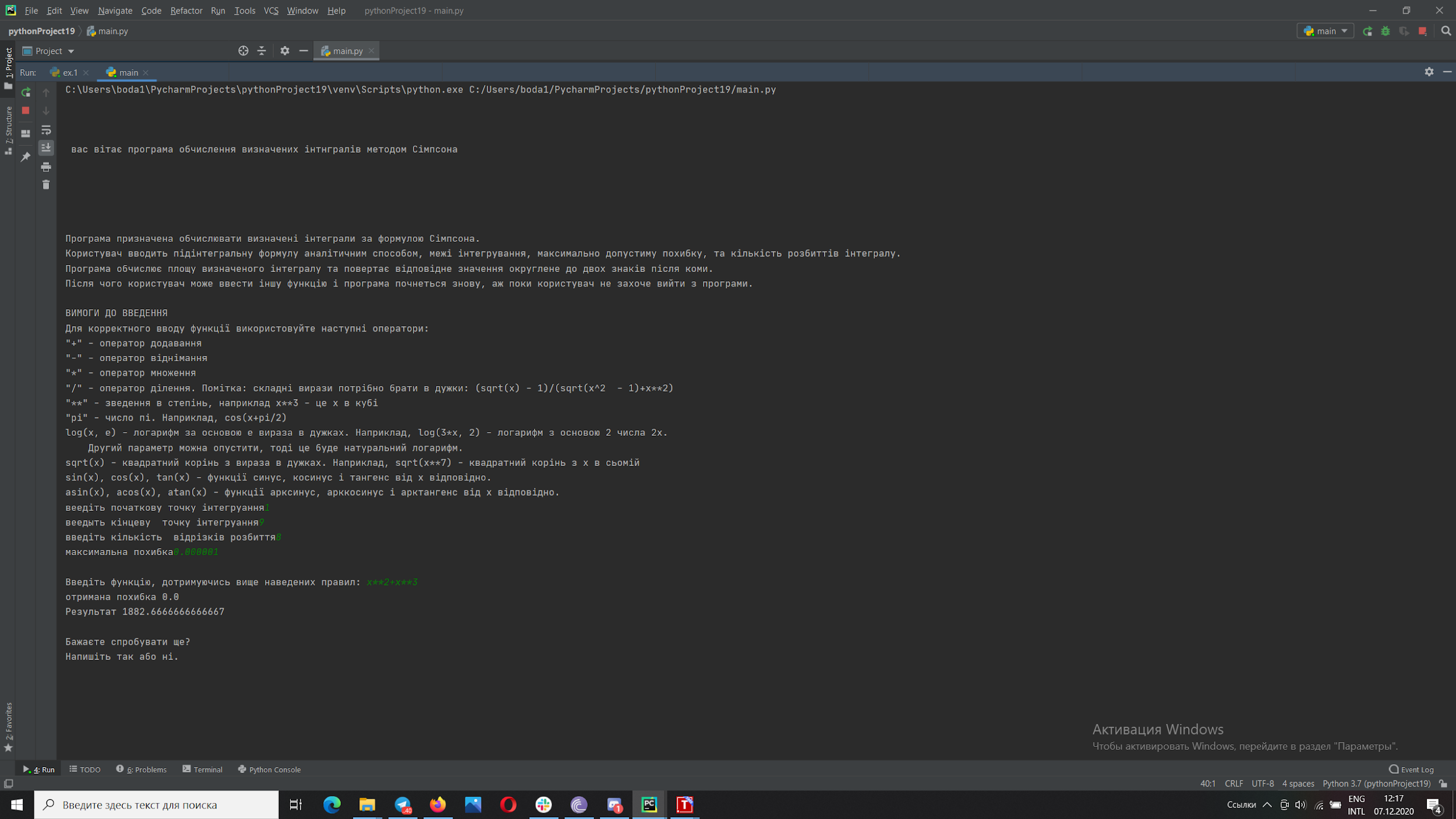


Рис-1.2