Павшук Ярослав Денисович, ФІТ 3-14, Варіант 21  
  
**Метод Трапецій**  
  
from scipy import integrate

import numpy as np

# Задаємо функцію, яку необхідно інтегрувати

def f(x):

return 1 / x\*\*2

# Задаємо межі інтегрування та початкову кількість розбиттів

a = 3.2

b = 4

n = 10

# Обчислюємо значення інтегралу методом трапецій

integral = integrate.trapz([f(xi) for xi in np.linspace(a, b, n)], dx=(b - a) / n)

# Виводимо результат

print("Метод трапецій:", round(integral, 4))

# Перевірка

v, err = integrate.quad(f, a, b)

print("Перевірка для методу трапецій:", round(v, 4))  
  
  
**Метод Сімпсона**  
  
from scipy import integrate

import numpy as np

# Задаємо функцію, яку необхідно інтегрувати

def f(x):

return (x + 1) \* np.sin(x)

# Задаємо межі інтегрування та початкову кількість розбиттів

a = 1.6

b = 2.4

n = 8

# Обчислюємо значення інтегралу методом Сімпсона

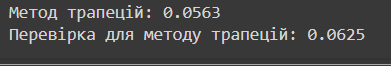
integral = integrate.simps([f(xi) for xi in np.linspace(a, b, n)], dx=(b - a) / (n - 1))

# Виводимо результат

print("Метод Сімпсона:", round(integral, 4))

# Перевірка

v, err = integrate.quad(f, a, b)

print("Перевірка для методу Сімпсона:", round(v, 4))  
  
  
  
**Метод Трикутника**  
  
from scipy import integrate

import numpy as np

# Задаємо функцію, яку необхідно інтегрувати

def f1(x):

return np.log10(x + 2) / x

# Задаємо межі інтегрування

a1 = 1.2

b1 = 2

# Обчислюємо значення інтегралу методом прямокутників (лівих) з точністю 0.0001 та умовою p=10

integral\_left = integrate.quad(f1, a1, b1, epsabs=1e-4, limit=10)[0]

# Виводимо результат

print("Метод прямокутників (лівих):", round(integral\_left, 5))  
  
