**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

Факультет физико-математических и естественных наук

ОТЧЕТ

по лабораторной работе № 3

дисциплина: Вычислительные методы

Студент:

Группа:

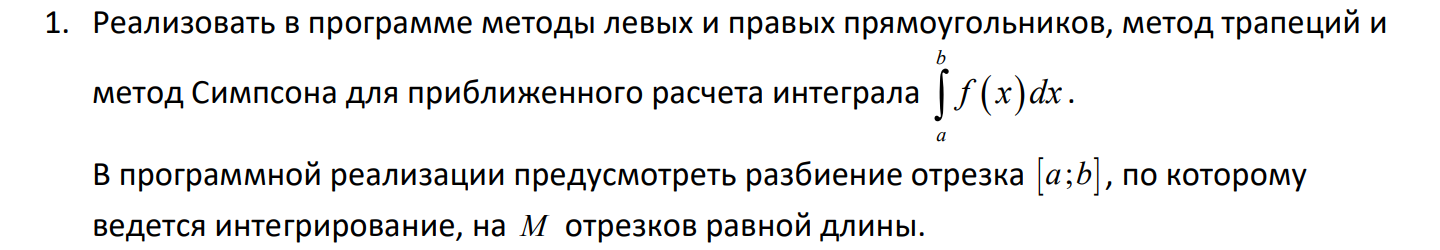
**МОСКВА**

2022 г.

**Мой вариант:**

****

**Ход работы:**



*def* \_rectangle\_rule(*func*, *a*, *b*, *nseg*, *frac*):

    dx = 1.0 \* (*b* - *a*) / *nseg*

    sum = 0.0

    xstart = *a* + *frac* \* dx

    for i in range(*nseg*):

        sum += *func*(xstart + i \* dx)

    return sum \* dx

# левыи прямоугольник

*def* left\_rectangle\_rule(*func*, *a*, *b*, *nseg*):

    return \_rectangle\_rule(*func*, *a*, *b*, *nseg*, 0.0)

# правыи прямоугольник

*def* right\_rectangle\_rule(*func*, *a*, *b*, *nseg*):

    return \_rectangle\_rule(*func*, *a*, *b*, *nseg*, 1.0)

*def* simpson\_rule(*func*, *a*, *b*, *nseg*):

    if (*nseg*%2 == 1):

*nseg* += 1

    dx = 1.0 \* (*b* - *a*) / *nseg*

    sum = (*func*(*a*) + 4 \* *func*(*a* + dx) + *func*(*b*))

    for i in range(1, int(*nseg* / 2)):

        sum += 2 \* *func*(*a* + (2 \* i) \* dx) + 4 \* *func*(*a* + (2 \* i + 1) \* dx)

    return sum \* dx / 3

*def* trapezoid\_rule(*func*, *a*, *b*, *nseg*):

    dx = 1.0 \* (*b* - *a*) / *nseg*

    sum = 0.5 \* (*func*(*a*) + *func*(*b*))

    for i in range(1, *nseg*):

        sum += *func*(*a* + i \* dx)

    return sum \* dx

# function:

f = *lambda* *x*: (*x*+1) \* math.sin(*x*)

x = symbols('x')

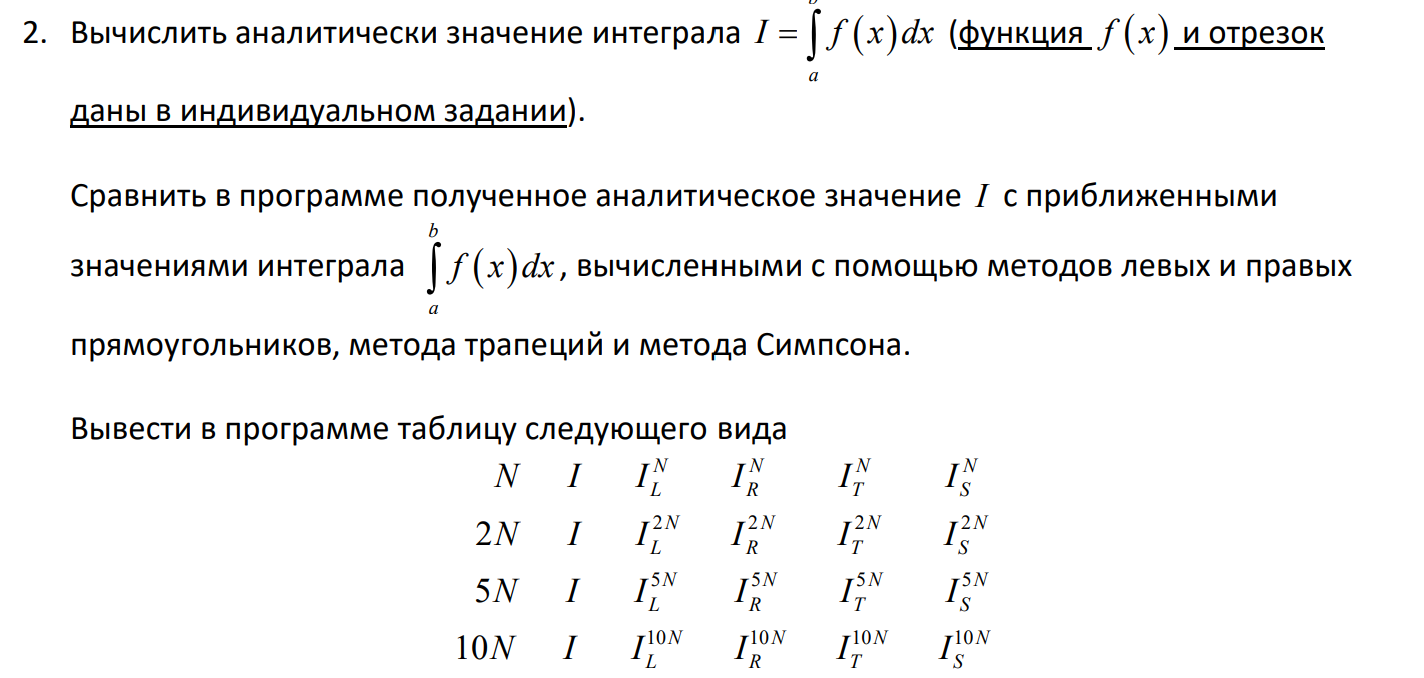
gf\_exp = (x+1) \* sin(x)

fI = integrate(gf\_exp,  (x, -1, 1))

numb = float(fI)

# N:

N = 16

****

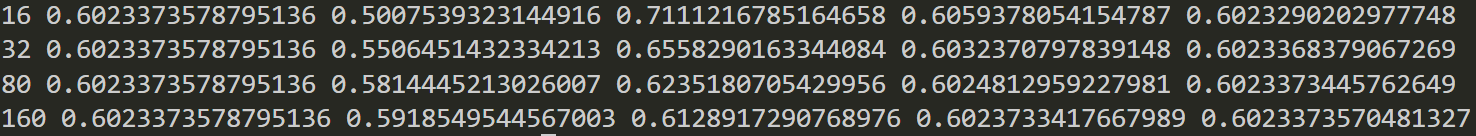
print(str(N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N)) + " " + str(right\_rectangle\_rule(f,-1,1,N)) + " " + str(trapezoid\_rule(f,-1,1,N)) + " " + str(simpson\_rule(f,-1,1,N)));

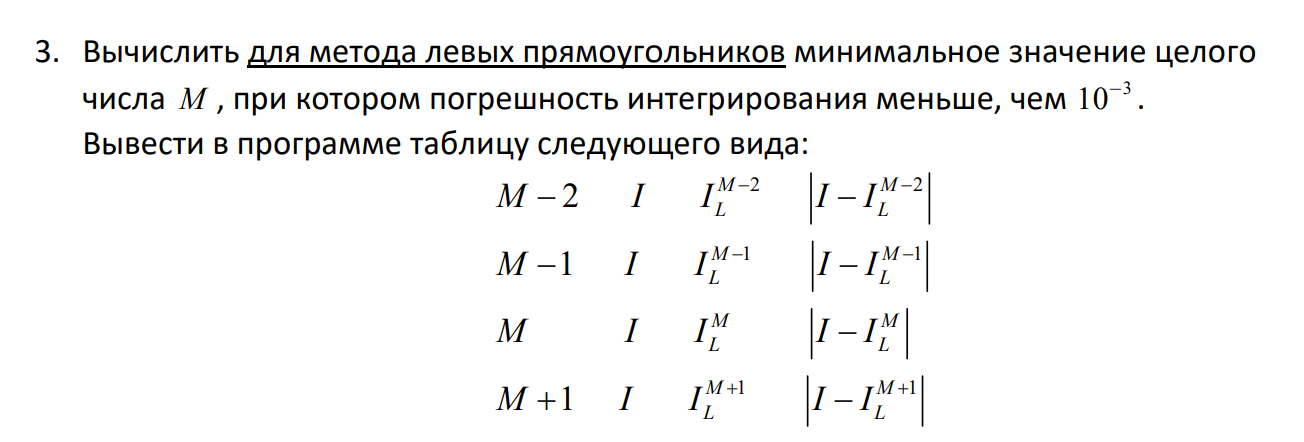
print(str(2\*N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,2\*N)) + " " + str(right\_rectangle\_rule(f,-1,1,2\*N)) + " " + str(trapezoid\_rule(f,-1,1,2\*N)) + " " + str(simpson\_rule(f,-1,1,2\*N)));

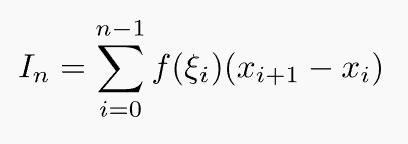
print(str(5\*N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,5\*N)) + " " + str(right\_rectangle\_rule(f,-1,1,5\*N)) + " " + str(trapezoid\_rule(f,-1,1,5\*N)) + " " + str(simpson\_rule(f,-1,1,5\*N)));

print(str(10\*N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,10\*N)) + " " + str(right\_rectangle\_rule(f,-1,1,10\*N)) + " " + str(trapezoid\_rule(f,-1,1,10\*N)) + " " + str(simpson\_rule(f,-1,1,10\*N)));

**Вывод:**

****

****

****

****

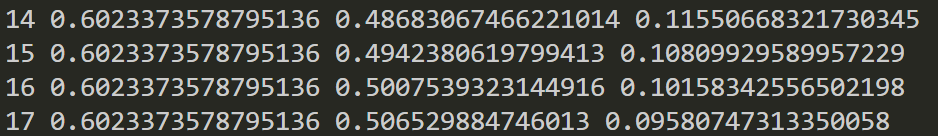
print(str(N-2) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N-2)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N-2))))

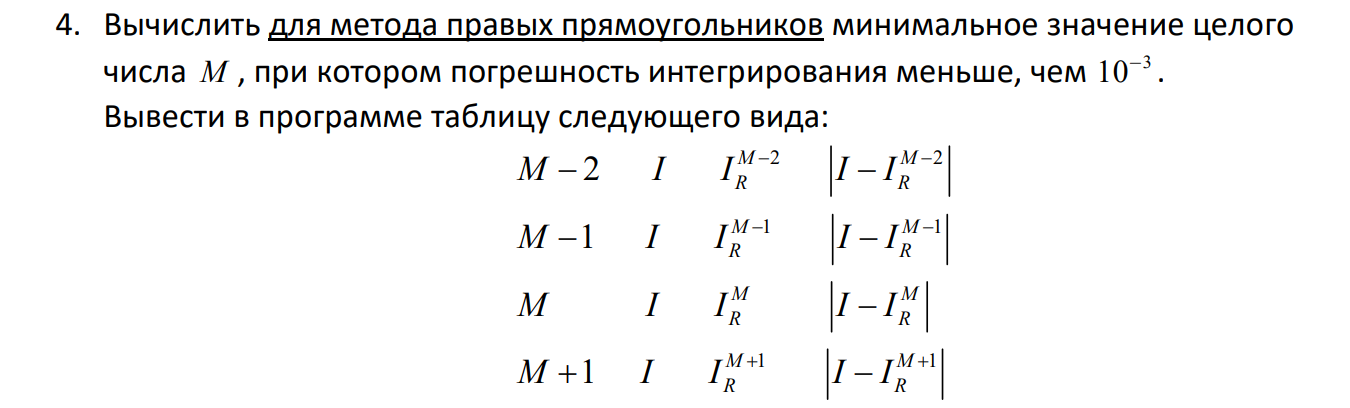
print(str(N-1) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N-1)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N-1))))

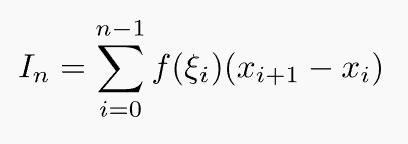
print(str(N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N+1)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N+1))))

**Вывод:**

****

****

****

****

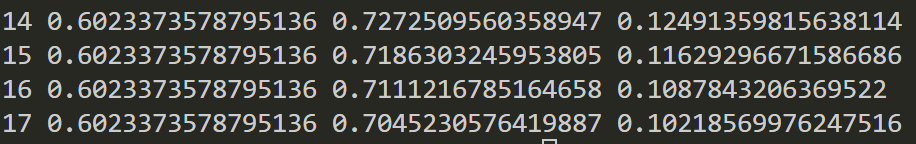
print(str(N-2) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N-2)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N-2))))

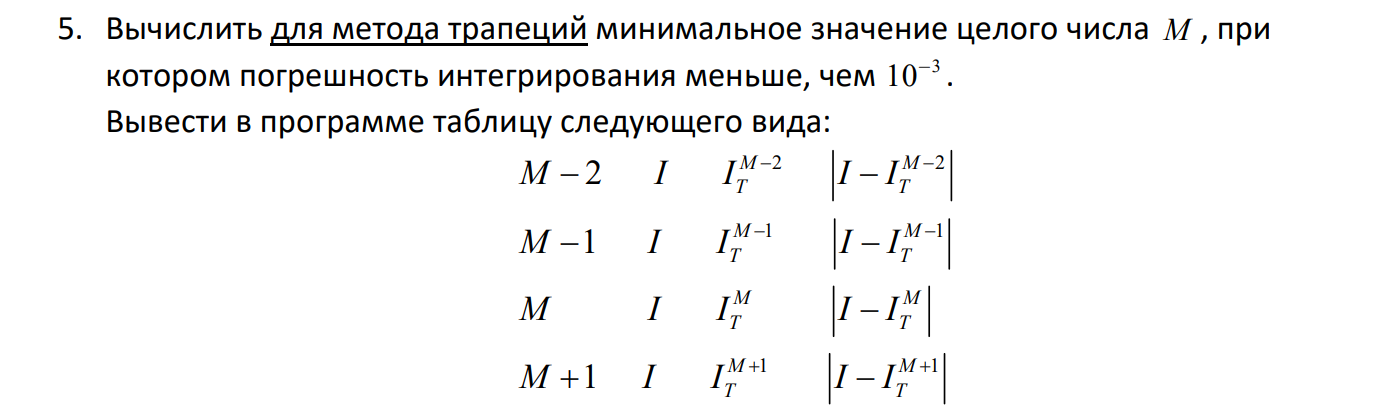
print(str(N-1) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N-1)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N-1))))

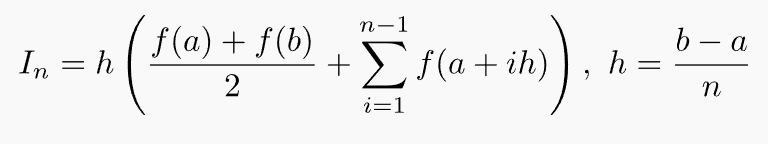
print(str(N) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N+1)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N+1))))

**Вывод:**

****

****

****

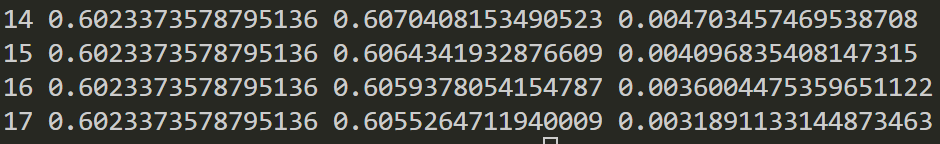
print(str(N-2) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N-2)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N-2))))

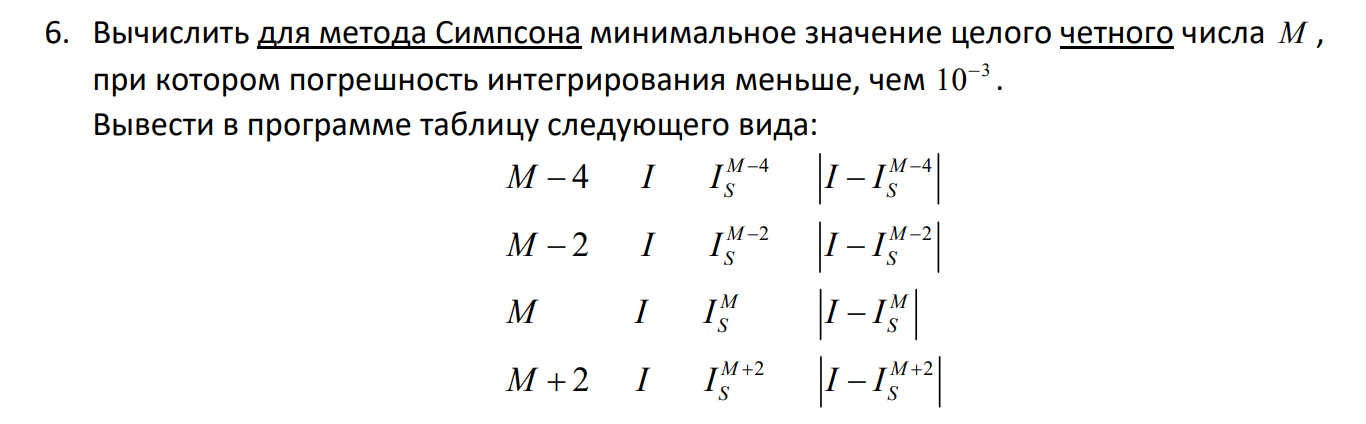
print(str(N-1) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N-1)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N-1))))

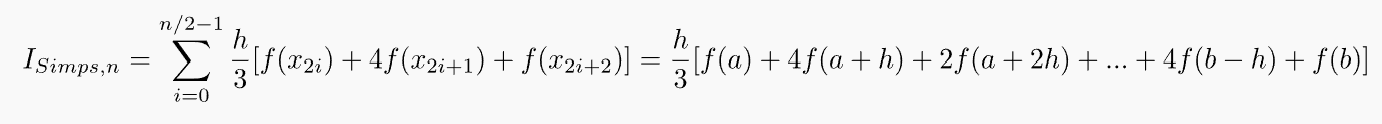
print(str(N) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N+1)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N+1))))

**Вывод:**

****

****

****

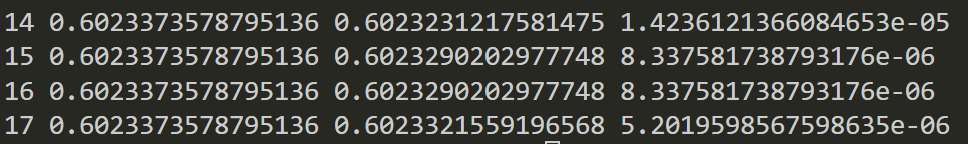
print(str(N-2) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N-2)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N-2))))

print(str(N-1) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N-1)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N-1))))

print(str(N) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N+1)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N+1))))

**Вывод:**

****

**КОД:**

import math

from sympy import \*

*def* \_rectangle\_rule(*func*, *a*, *b*, *nseg*, *frac*):

    dx = 1.0 \* (*b* - *a*) / *nseg*

    sum = 0.0

    xstart = *a* + *frac* \* dx

    for i in range(*nseg*):

        sum += *func*(xstart + i \* dx)

    return sum \* dx

# левыи прямоугольник

*def* left\_rectangle\_rule(*func*, *a*, *b*, *nseg*):

    return \_rectangle\_rule(*func*, *a*, *b*, *nseg*, 0.0)

# правыи прямоугольник

*def* right\_rectangle\_rule(*func*, *a*, *b*, *nseg*):

    return \_rectangle\_rule(*func*, *a*, *b*, *nseg*, 1.0)

*def* simpson\_rule(*func*, *a*, *b*, *nseg*):

    if (*nseg*%2 == 1):

*nseg* += 1

    dx = 1.0 \* (*b* - *a*) / *nseg*

    sum = (*func*(*a*) + 4 \* *func*(*a* + dx) + *func*(*b*))

    for i in range(1, int(*nseg* / 2)):

        sum += 2 \* *func*(*a* + (2 \* i) \* dx) + 4 \* *func*(*a* + (2 \* i + 1) \* dx)

    return sum \* dx / 3

*def* trapezoid\_rule(*func*, *a*, *b*, *nseg*):

    dx = 1.0 \* (*b* - *a*) / *nseg*

    sum = 0.5 \* (*func*(*a*) + *func*(*b*))

    for i in range(1, *nseg*):

        sum += *func*(*a* + i \* dx)

    return sum \* dx

# function:

f = *lambda* *x*: (*x*+1) \* math.sin(*x*)

x = symbols('x')

gf\_exp = (x+1) \* sin(x)

fI = integrate(gf\_exp,  (x, -1, 1))

numb = float(fI)

# N:

N = 16

# outputs 1:

print(str(N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N)) + " " + str(right\_rectangle\_rule(f,-1,1,N)) + " " + str(trapezoid\_rule(f,-1,1,N)) + " " + str(simpson\_rule(f,-1,1,N)));

print(str(2\*N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,2\*N)) + " " + str(right\_rectangle\_rule(f,-1,1,2\*N)) + " " + str(trapezoid\_rule(f,-1,1,2\*N)) + " " + str(simpson\_rule(f,-1,1,2\*N)));

print(str(5\*N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,5\*N)) + " " + str(right\_rectangle\_rule(f,-1,1,5\*N)) + " " + str(trapezoid\_rule(f,-1,1,5\*N)) + " " + str(simpson\_rule(f,-1,1,5\*N)));

print(str(10\*N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,10\*N)) + " " + str(right\_rectangle\_rule(f,-1,1,10\*N)) + " " + str(trapezoid\_rule(f,-1,1,10\*N)) + " " + str(simpson\_rule(f,-1,1,10\*N)));

# output 2 left:

print("")

print(str(N-2) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N-2)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N-2))))

print(str(N-1) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N-1)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N-1))))

print(str(N) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(left\_rectangle\_rule(f,-1,1,N+1)) + " " + str(abs(numb - left\_rectangle\_rule(f,-1,1,N+1))))

# output 3 right:

print("")

print(str(N-2) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N-2)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N-2))))

print(str(N-1) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N-1)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N-1))))

print(str(N) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(right\_rectangle\_rule(f,-1,1,N+1)) + " " + str(abs(numb - right\_rectangle\_rule(f,-1,1,N+1))))

# output 4 trap:

print("")

print(str(N-2) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N-2)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N-2))))

print(str(N-1) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N-1)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N-1))))

print(str(N) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(trapezoid\_rule(f,-1,1,N+1)) + " " + str(abs(numb - trapezoid\_rule(f,-1,1,N+1))))

# output 5 simpson:

print("")

print(str(N-2) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N-2)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N-2))))

print(str(N-1) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N-1)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N-1))))

print(str(N) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N))))

print(str(N+1) + " " + str(numb) + " " + str(simpson\_rule(f,-1,1,N+1)) + " " + str(abs(numb - simpson\_rule(f,-1,1,N+1))))