

```

import scipy as sp
import numpy as np
from numpy.polynomial import polynomial as P

def intTrapecio( f, a, b, n ):
    if(n==1):
        resultado = (f(b) + f(a))*(b-a)/2
    else:
        h = (b-a)/n
        x=[ a+h*i for i in range(n+1)]
        resultado = ( h * ( f(x[0]) + f(x[n]) )/2 + sum([f(x[i+1]) for i in range(n-1)]) )

    return resultado

#Rk = R_{k,j-1} RK1 = R_{k-1,j-1}
def rombergParcial( Rk, Rk1, j ):
    return Rk + (1/(4**j-1))*(Rk - Rk1)

def romberg(f, a, b, k):
    R=[]
    R.append( intTrapecio( f, a, b, 1 ) )
    for i in range(k):
        R.append( intTrapecio( f, a, b, 2**(i+1) ) )

    for i in range(k):
        for j in range(k-i):
            R[j]=rombergParcial( R[j+1], R[j], j+1 )

    return R[0]

f = lambda x: np.log(x)
a=1
b=2
n=10

resultado = romberg( f, a, b, n )
print("\nEl valor aproximado es:",resultado)

```