```python
"""
Registration: xxxx;
Description: Dirac Delta, Gaussian Integral, Convolution
Author: AKB
"""

import numpy as np
import scipy.integrate as sci
import matplotlib.pyplot as plt
from scipy.signal import gaussian

# Logical case switch for different problems to choose from
prob1=1; prob2=1; prob3=1; prob4=1; prob5=0;

# ===== Improper integral for ddelta ===== #
if(prob1):
   def f(x,mu,sig):
       return np.exp(-(x-mu)**2/(2.0*sig**2))*(x+3)/np.sqrt(2.0*np.pi*sig**2)  #[-
np.inf,np.inf]

   # Enter standard deviation and integration limits
   #mu, sig, low, up = input('Enter sigma & integration limits : ')
   mu = 2.0; sig = 0.015; low = -np.inf; up = np.inf;

   # Eradicating the Crossover problem by narrowing the bracket
   low = mu - 10*sig; up = mu + 10*sig;

   # Peform the infinite interval integral using Quadrature
   I, err = sci.quad(f,low,up,args=(mu,sig))

   # Print result
   print ('Integral computed value = ', I, ' with error = ', err)

# ====== Gaussian Integral ====== #
if(prob2):
   def f(x,a,b,c): return np.exp(-a*x**2 + b*x + c) #[-np.inf,np.inf]

   # Enter the coefficients and integral bounds
   #a,b,c,low,up = input('Enter a,b,c coefficients & integration limits : ')
   a = 1; b = 2; c = 1; low = -np.inf; up = np.inf;

   # Peform indefinite integral using Quadrature
   I_num, err = sci.quad(f, low, up, args=(a,b,c))
   I_theo     = np.sqrt(np.pi/a)*np.exp(b**2/(4.0*a)+c)

   # Print/compare results
   print ('Integral_',low,'^',up,' e^(-',a,'x^2+',b,'x+',c,') dx = ', I_num)
   print ('Theoretical value of the Integral = ', I_theo)
   print ('Absolute error = ', err, ', Relative error = ', I_num - I_theo)

# ====== Convolution of two Gaussian ====== #
if(prob3):

   def gauss(x,mu,sig):
       return np.exp(-((x-mu)**2.0)/(2.0*sig**2.0))/np.sqrt(2.0*np.pi)/sig

   # Choose two normal distributions to convolve
   mu1 = 0; sig1 = 0.3
   mu2 = 0; sig2 = 0.2;
   x = np.linspace(-2, 2, 500)
   dx = x[1] - x[0]
   convolution = np.convolve(gauss(x,mu1,sig1), gauss(x,mu2,sig2), mode="same")*dx
   #sigc = np.sqrt(sig1**2 * sig2**2/(sig1**2 + sig2**2)) # product std
   #ampc = sigc/(np.sqrt(2*np.pi)*sig1*sig2)              # product amplitude
   sigc = np.sqrt(sig1**2 + sig2**2)              # convolution std
   ampc = 1.0/np.sqrt(2*np.pi*(sig1**2 + sig2**2)) # convolution amplitude
```

```
    # Plot
    plt.figure()
    plt.plot(x, gauss(x,mu1,sig1), '--b+', lw=.4, label=r"$\mathcal{N}
_1("+str(mu1)+","+str(sig1)+")$")
    plt.plot(x, gauss(x,mu2,sig2), '--g<', lw=.4, label=r"$\mathcal{N}
_2("+str(mu2)+","+str(sig2)+")$")
    plt.plot(x, convolution,       '--rx', lw=.4, label=r"$\mathcal{N}
_c("+str(mu1+mu2)+","+str(round(sigc,2))+")$")
    plt.title("(Convolution) Amplitude="+str(round(ampc,2)))
    plt.legend(loc='best', prop={'size':18})
    plt.xlabel("x", size=16)
    plt.xticks(size = 14)
    plt.ylabel(r"$\mathcal{N}(\mu,\sigma)$", size=16)
    plt.yticks(size = 14)
    plt.grid()
    plt.tight_layout()
    #plt.savefig('plot/02_convol.pdf')
    plt.show()

# ====== Int_(x1-a)^(x2+a) f(x) ddelta(x-a) dx = f(a) ====== #
if(prob4):
    from sympy import *
    def f(x) :
        #return x**2
        #return exp(-x**2+x+1)
        return sin(x)

    # Enter integration limits x1, x2 and common additive a
    #x1, x2, a = input('Enter integration limits x1, x2 and common additive a : ')
    x1 = 1.0; x2 = 1.5; a = 1.0;

    # Use symbolic computation to perform the definite integral
    x = Symbol('x')
    I = integrate(f(x)*DiracDelta(x-a), (x, a-x1, a+x2))

    # Print result to get value by evalf
    print ('Integral_(',a,'-',x1,')^(',a,'+',x2,') x^2 ddelta(x-',a,')dx = ', I.evalf())

# ===== Improper integral for ddelta ===== #
if(prob5):
    from sympy import *
    import numpy as np

    def f(x):
        return x**2

    def ddelta(x,a):
        eps = 1.0;
        #return exp(-abs(x-a)/eps)/(2.0*eps)
        return exp(-(x-a)**2/(4.0*eps))/(sqrt(4.0*pi*eps))

    # Enter integration limits x1, x2 and common additive a
    #x1, x2, a = input('Enter integration limits x1, x2 and common additive a : ')
    x1 = 1.0; x2 = 1.5; a = 1.0;

    # Use symbolic computation to perform the definite integral
    x = Symbol('x')
    I = integrate(f(x)*ddelta(x,a), (x, a-x1, a+x2))

    print ('Integral_(',a,'-',x1,')^(',a,'+',x2,') x^2 ddelta(x-',a,')dx = ', I.evalf())

"""
########### Results (prob1):
a) Enter sigma & integration limits : 10.0, -np.inf, np.inf
Integral computed value =  5.0  with error =  3.21787212976e-08
```

```
b) Enter sigma & integration limits : 1.0,-np.inf,np.inf
Integral computed value =  5.0 with error =  2.78199541533e-08

c) Enter sigma & integration limits : 0.1,-np.inf,np.inf
Integral computed value =  5.0 with error =  1.38704512181e-08

d) Enter sigma & integration limits : 0.05, -np.inf, np.inf
Integral computed value =  5.0  with error =  2.63478332566e-08

e) Enter sigma & integration limits : 0.025, -np.inf, np.inf
Integral computed value =  5.0  with error =  1.11929395302e-08  #### CROSSOVER ####

f) Enter sigma & integration limits : 0.01,-np.inf,np.inf
Integral computed value =  7.57910251848e-51 with error =  1.37155184913e-50

g) After redefining the limit [mu-10*sig, mu+10*sig] to compute delta
Integral computed value =  5.0  with error =  4.3355779296e-09

########### Results (prob2):
Enter a,b,c coefficients & integration limits : 1,2,1,-np.inf,np.inf
Integral_ -inf ^ inf  e^(- 1 x^2+ 2 x+ 1 ) dx =  13.0967609371
Theoretical value of the Integral =  13.0967609371
Absolute error =  2.7105554618e-10 , Relative error =  0.0

########### Results (prob4):
Integral_( 5.0 - 1.0 )^( 5.0 + 1.5 ) x^2 ddelta(x- 5.0 )dx =  25.0000000000000
Integral_( -5.0 - 1.0 )^( -5.0 + 1.5 ) x^2 ddelta(x- -5.0 )dx =  25.0000000000000

Integral_( 5.0 - 1.0 )^( 5.0 + 1.5 ) exp(-x**2+x+1) ddelta(x- 5.0 )dx =
5.60279643753727e-9
Integral_( -5.0 - 1.0 )^( -5.0 + 1.5 ) exp(-x**2+x+1) ddelta(x- -5.0 )dx =
2.54366564737692e-13

Integral_( 5.0 - 1.0 )^( 5.0 + 1.5 ) sin(x) ddelta(x- 5.0 )dx =  -0.958924274663138
Integral_( -5.0 - 1.0 )^( -5.0 + 1.5 ) sin(x) ddelta(x- -5.0 )dx =  0.958924274663138
"""
```