```python
"""
Registration: xxxx;
Description: Lagrange Interpolation Method
Author: AKB
"""
import numpy as np
from scipy.interpolate import lagrange
import matplotlib.pyplot as plt

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

if(prob1):
   # Enter the table y = f(x) and evaluation point xx
   n = int(input('Enter the number of terms in the table: '))
   print ('Populate the x and y arrays: \n')
   x = np.array([float(input("x"+str(i)+" : ")) for i in range(n)])
   y = np.array([float(input("y"+str(i)+" : ")) for i in range(n)])
   print ('x coordinates: ',x)
   print ('y coordinates: ',y)
   xx = float(input('Enter point at which the polynomial is to be evaluated : \n'))

elif(prob2):
   # Enter the x coordinate and evaluation point xx
   n = int(input('Enter the number of terms in the abcissa: '))
   print ('Populate x array: \n')
   x = np.array([eval(input("x"+str(i)+" : ")) for i in range(n)]);
   y = np.sin(x)
   print ('x coordinates: ',x)
   xx = float(input('Enter point at which the polynomial is to be evaluated : \n'))

# Do the interpolation & print the results
k = 0.0
for i in range(n):
    s=t=1
    for j in range(n):
       if j!=i:
         s = s*(xx - x[j]);
         t = t*(x[i] - x[j]);
    k += (s/t)*y[i];

print ('Corresponding value of the variable y is', k)

# Plot
plt.figure(1)
plt.plot(x, y, 'ro', ms=8, label=r"$Data points$") # plot points
lim1=0; lim2=1; tmp = np.linspace(lim1, lim2, len(x)) # auxiliary array
polyx = lagrange(tmp, x) # construct poly1d x & y polynomials
polyy = lagrange(tmp, y)
tmp = np.linspace(lim1, lim2, 100)        # auxiliary array
interpx = polyx(tmp)
interpy = polyy(tmp)
plt.plot(interpx, interpy,'k.',label = r"$P(x)$")
plt.grid()
plt.xlabel(r'$x$', size=16);
plt.xticks(size=14)
plt.ylabel(r'$y=f(x), P(x)$', size=16);
plt.yticks(size=14)
plt.text(1.0,0.42,'$n=%s,points=100$'%(n),size=20)
plt.legend(loc='best', prop={'size':16})
plt.title(r'Lagrange Interpolation', size=16);
#plt.savefig('plot/06_lagrange.pdf')
plt.show()

"""
Results :
```

```
Enter the number of terms in the table: 6
Populate the x and y arrays:
x0 : .1
x1 : .2
x2 : .3
x3 : .4
x4 : .5
x5 : .6
y0 : .545
y1 : .331
y2 : .275
y3 : .348
y4 : .240
y5 : .235
x coordinates:  [ 0.1  0.2  0.3  0.4  0.5  0.6]
y coordinates:  [ 0.545  0.331  0.275  0.348  0.24   0.235]
Enter point at which the polynomial is to be evaluated : .25
Corresponding value of the variable y is 0.26822265625
=============================================================================
Enter the number of terms in the abcissa: 5
Populate x array:
x0 : 0
x1 : np.pi/4
x2 : np.pi/2
x3 : 3*np.pi/4
x4 : np.pi
x coordinates:  [ 0. 0.78539816  1.57079633  2.35619449  3.14159265]
Enter point at which the polynomial is to be evaluated :    0.5
Corresponding value of the variable y is 0.4783926678401498
"""
```