```
"""
Registration: xxxx;
Description: Newton-Gregory Forward Interpolation Method
Author: AKB
"""

import numpy as np

# Enter the table y = f(x) and give x
n = int(input('Enter the number of table entries equal to degree of polynomial : '))
print ('Input the x coordinate : \n')
x = np.array([eval(input("x"+str(i)+" : ")) for i in range(n)])
y = np.sin(x)
xx = float(input('Enter point at which the polynomial is to be evaluated : \n'))
a = np.zeros(n)  # null-assign the a cofficients

# Compute the divided difference coefficients
a[0] = y[0]
for k in range(0,n):
    w = 1
    p = 0
    for j in range(0,k):
        p += a[j]*w
        w = w*(x[k] - x[j])
    a[k] = (y[k]-p)/w

print ('Divided difference coefficients are ', a)

# Compute the polynomial at given value

px = a[n-1] # polynomial value at xx
for k in range(n-2,-1,-1):
    xd = xx - x[k]
    px = a[k] + px*xd

print ('Corresponding value of y is ', px)

"""
Results :
Enter the number of table entries equal to degree of polynomial : 5
Input the x coordinate :
x0 : 0
x1 : np.pi/4
x2 : np.pi/2
x3 : 3.0*np.pi/4
x4 : np.pi
Enter point at which the polynomial is to be evaluated : 0.5
Divided difference coefficients are  [ 0. 0.90031632 -0.33574887 -0.05902388  0.03757577]
Corresponding value of y is  0.4783926678401497
"""
```