BACKWARD INTERPOLATION

```
import numpy as np
import math

x = [0.2, 0.22, 0.24, 0.26, 0.28, 0.3]
y = [1.6595, 1.6698, 1.6804, 1.6912, 1.7024, 1.7139]

n = len(x)

diff_table = np.zeros([n, n])
#print(diff_table)

diff_table[0] = y

for i in range(1, n):
    for j in range(i, n):
        diff_table[i][j] = round(diff_table[i-1][j] - diff_table[i-1][j-
1], 4)


#print(diff_table)
Y = np.transpose(diff_table)

print("Backward Difference Table")
print(Y)

# For y(0.29)

xn = 0.3
x = 0.29
yn = y[n-1]
h = 0.02

p = (x - xn) / h

y_x = yn
for i in range(n - 5):
    P = 1
    for j in range(i + 1):
        P = P * (p + j)

    y_x = y_x + (P * Y[1][i + 1]) / math.factorial(i + 1)

print("\n y(0.29) = ", y_x)
```

OUTPUT –

```
Backward Difference Table:

[[ 1.6595]
 [ 1.6698  1.0300e-02]
 [ 1.6804  1.0600e-02  3.0000e-04]
 [ 1.6912  1.0800e-02  2.0000e-04 -1.0000e-04]
 [ 1.7024  1.1200e-02  4.0000e-04  2.0000e-04  3.0000e-04]
 [ 1.7139  1.1500e-02  3.0000e-04 -1.0000e-04 -3.0000e-04 -6.0000e-04]]

 y(0.29) =  1.70875
```