Newton's Backward difference.py

```
#-------------------------------------------------------------------------------
# Name: Newton's Backward difference
# Purpose: Implemention of Newton's Backward difference interpolation
#
# Author: Abdul Qadir Lonara
#
# Created:
# Copyright: (c) Abdul Qadir Lonara 2023
# Licence: <your licence>
#-------------------------------------------------------------------------------
import math
#-------Input-------------
# creating an empty list
x = []
# number of elements as input
n = int(input("Enter number of elements : "))
print("Enter x cordinates : ")
# iterating till the range
for i in range(0, n):
ele = float(input("X"+str(i)+" : "))
x.append(ele) # adding the element
print(x)
y = []
print("Enter y cordinates : ")
# iterating till the range
for i in range(0, n):
ele = float(input("Y"+str(i)+" : "))
y.append(ele) # adding the element
print(y)
xi=float(input("Enter the point at which to evaluate the polynomial : "))
# create a table to hold the backward difference values
bwd_diff_table = [[0] * n for i in range(n)]
# fill in the first column with the y-values
for i in range(n):
bwd_diff_table[i][0] = y[i]
# compute the backward difference values
for j in range(1, n):
```

for i in range(n - j):

bwd_diff_table[i][j] = (bwd_diff_table[i][j - 1] - bwd_diff_table[i + 1][j - 1])

use the backward difference values to interpolate the polynomial

yi = bwd_diff_table[n - 1][0]

prod = 1

for j in range(1, n):

prod *= (xi - x[n - j])

yi += (prod * bwd_diff_table[n - j - 1][j]) / math.factorial(j)

print("The value of the polynomial at x="+str(xi)+" is "+str(yi))

Output :-

Enter number of elements : 6

Enter x cordinates :

X0 : 0.1

X1 : 0.3

X2 : 0.5

X3 : 0.7

X4 : 0.9

X5 : 1.1

[0.1, 0.3, 0.5, 0.7, 0.9, 1.1]

Enter y cordinates :

Y0 : -1.699

Y1 : -1.073

Y2 : -0.375

Y3 : 0.443

Y4 : 1.429

Y5 : 2.631

[-1.699, -1.073, -0.375, 0.443, 1.429, 2.631]

Enter the point at which to evaluate the polynomial : 1

The value of the polynomial at x=1.0 is 2.0501439999999997