NEWTON FORWARD DIFFERENCE INTERPOLATION IMPLEMENTATION IN PYTHON

SUBMITTED BY: UMANG KANCHAN

SUBMITTED TO: Dr. AYESHA CHOUDHARY

ASSIGNMENT 6A

NUMERICAL METHODS (CS-406)

2023

def divided\_diff(*x*, *y*):

    n = len(x)

    coefficients = []

    for i in range(n):

        coefficients.append(y[i])

    for j in range(1, n):

        for i in range(n-1, j-1, -1):

            coefficients[i] = (coefficients[i] - coefficients[i-1]) / (x[i] - x[i-j])

    return coefficients

def newton\_interpolation(*x*, *y*, *xi*):

    coefficients = divided\_diff(x, y)

    n = len(x)

    result = coefficients[n-1]

    for i in range(n-2, -1, -1):

        result = result \* (xi - x[i]) + coefficients[i]

    return result

# user input

n = *int*(input("Enter the number of data points: "))

x = []

y = []

for i in range(n):

    x\_i = *float*(input(f"Enter x{i+1}: "))

    y\_i = *float*(input(f"Enter y{i+1}: "))

    x.append(x\_i)

    y.append(y\_i)

# test

xi = *float*(input("Enter the value of xi for interpolation: "))

yi = newton\_interpolation(x, y, xi)

print(f"Interpolated value at xi = {xi} is {yi}")

SAMPLE OUTPUT

Enter the number of data points: 3

Enter x1: 0

Enter y1: 1

Enter x2: 1

Enter y2: 2

Enter x3: 2

Enter y3: 3

Enter the value of xi for interpolation: 0

Interpolated value at xi = 0.0 is 1.0