NEWTON BACKWARD DIFFERENCE INTERPOLATION IMPLEMENTATION IN PYTHON

SUBMITTED BY: UMANG KANCHAN

SUBMITTED TO: Dr. AYESHA CHOUDHARY

ASSIGNMENT 6B

NUMERICAL METHODS (CS-406)

2023

# function to calculate the divided difference table

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

    for i in range(1, n):

        for j in range(n - 1, i - 1, -1):

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

    return y

# function to perform Newton's Backward Interpolation

def newton\_backward\_interpolation(*x*, *y*, *n*, *value*):

    # calculate divided difference table

    div\_diff\_table = divided\_diff\_table(x, y, n)

    # initialize the result with the last term of the table

    result = div\_diff\_table[n - 1]

    # calculate the interpolation polynomial

    for i in range(n - 2, -1, -1):

        result = result \* (value - x[i]) + div\_diff\_table[i]

    return result

# main function to take user input and perform interpolation

def main():

    # take user input for number of data points

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

    # initialize x and y arrays

    x = []

    y = []

    # take user input for x and y values

    for i in range(n):

        x\_val = *float*(input(f"Enter x[{i}]: "))

        y\_val = *float*(input(f"Enter y[{i}]: "))

        x.append(x\_val)

        y.append(y\_val)

    # take user input for value to interpolate

    value = *float*(input("Enter the value to interpolate: "))

    # perform interpolation and print result

    result = newton\_backward\_interpolation(x, y, n, value)

    print(f"Interpolated value at {value}: {result}")

# call the main function

if \_\_name\_\_ == "\_\_main\_\_":

    main()

SAMPLE OUTPUT

Enter the number of data points: 3

Enter x[0]: 0

Enter y[0]: 1

Enter x[1]: 1

Enter y[1]: 2

Enter x[2]: 2

Enter y[2]: 3

Enter the value to interpolate: 9

Interpolated value at 9.0: 10.0