Janvi Chauhan

Rollno:-03

Conm Practical Assignment 2

1. Write a program to find the value of f (0.4) using Lagrange's interpolation formula for the table of values given below.

x 0.3 0.5 0.6

f(x) 0.61 0.69 0.72

```
def lagrange_interpolation(x_values, y_values, x):
    result = 0
    n = len(x_values)

    for i in range(n):
        term = y_values[i]
        for j in range(n):
            if i != j:
                term *= (x - x_values[j]) / (x_values[i] - x_values[j])
        result += term

    return result
```

```
x_values = [0.3, 0.5, 0.6]
y_values = [0.61, 0.69, 0.72]

x = 0.4
result = lagrange_interpolation(x_values, y_values, x)
print(f"f(0.4) using Lagrange's interpolation is: {result:.4f}")
```

2. Write a program to find the value of f (2) using Newton's Divided Difference interpolation

formula for the table of values given below.

x -1 0 3 6 7

f(x) 3 -6 39 822 1611

```
def divided_diff_table(x_values, y_values, n):
    diff_table = [[0 for i in range(n)] for j in range(n)]

    for i in range(n):
        diff_table[i][0] = y_values[i]

    for j in range(1, n):
        for i in range(n - j):
            diff_table[i][j] = (diff_table[i+1][j-1] - diff_table[i][j-1]) / (x_values[i+j] - x_values[i])
```

```
    return diff_table


def newton_divided_diff(x_values, y_values, x):
    n = len(x_values)
    diff_table = divided_diff_table(x_values, y_values, n)
    result = diff_table[0][0]
    product = 1

    for i in range(1, n):
        product *= (x - x_values[i-1])
        result += diff_table[0][i] * product

    return result

# Table values
x_values = [-1, 0, 3, 6, 7]
y_values = [3, -6, 39, 822, 1611]

# Finding f(2)
x = 2
result = newton_divided_diff(x_values, y_values, x)
print(f"f(2) using Newton's Divided Difference interpolation is: {result:.4f}")
```

3. Write a program to find the value of f(0.5) by using Forward Difference interpolation on

the following table of data.

x 0 1 2 3 4

f(x) 1 4 16 64 256

```
def forward_diff_table(y_values, n):
    diff_table = [y_values]

    for i in range(1, n):
        new_row = [diff_table[-1][j+1] - diff_table[-1][j] for j in range(n-i)]
        diff_table.append(new_row)

    return diff_table

def forward_difference(x_values, y_values, x):
    n = len(x_values)
    h = x_values[1] - x_values[0]
    diff_table = forward_diff_table(y_values, n)

    # Starting value of the function
    result = y_values[0]
    u = (x - x_values[0]) / h
    u_term = u

    # Iterating through the forward difference table
    for i in range(1, n):
        result += u_term * diff_table[i][0] / factorial(i)
```

```
        u_term *= (u - i)

    return result

def factorial(n):
    return 1 if n == 0 else n * factorial(n-1)

x_values = [0, 1, 2, 3, 4]
y_values = [1, 4, 16, 64, 256]

x = 0.5
result = forward_difference(x_values, y_values, x)
print(f"f(0.5) using Forward Difference interpolation is: {result:.4f}")
```

4. Write a program to find the value of f(7.5) by using Backward Difference interpolation on the following table of data.

x 3 4 5 6 7 8

f(x) 28 65 126 217 344 513

```
def backward_diff_table(y_values, n):
    diff_table = [y_values]

    for i in range(1, n):
        new_row = [diff_table[-1][j] - diff_table[-1][j-1] for j in range(1, n-i+1)]
```

```python
        diff_table.append(new_row)

    return diff_table

def backward_difference(x_values, y_values, x):
    n = len(x_values)
    h = x_values[1] - x_values[0]
    diff_table = backward_diff_table(y_values, n)

    # Starting value of the function
    result = y_values[-1]
    u = (x - x_values[-1]) / h
    u_term = u

    # Iterating through the backward difference table
    for i in range(1, n):
        result += u_term * diff_table[i][-1] / factorial(i)
        u_term *= (u + i)

    return result

def factorial(n):
    return 1 if n == 0 else n * factorial(n-1)

# Table values
x_values = [3, 4, 5, 6, 7, 8]
```

```
y_values = [28, 65, 126, 217, 344, 513]


x = 7.5
result = backward_difference(x_values, y_values, x)
print(f"f(7.5) using Backward Difference interpolation is: {result:.4f}")
```