Experiment 8

Aim

Interpolation using Newton's forward formula and Newton's backward formula.

Algorithm

1. Start
2. Enter the no. of data values 'n'.
3. Declare the array 'x' of size n and y[][] for xx values and and difference table, declare the variable i,j.
4. Calculate the difference table between corresponding values of y for values of x by looping.

For Forward Difference table use `y[j][i] = y[j + 1][i - 1] - y[j][i - 1];`

And for Backward Difference table use `y[j][i] = y[j][i - 1] - y[j - 1][i - 1]`

5. Display the tables
6. Stop

Code (FFORWARD DIFFERENCE)

```
// Newton forward interpolation
#include <bits/stdc++.h>
using namespace std;


int main()
{
    // Number of values given
```

```
    int n = 6;
    float x[] = { 15, 20, 25, 30, 35, 40 };

    // y[][] is used for difference table
    // with y[][0] used for input
    float y[n][n];
    y[0][0] = 0.2588190;
    y[1][0] = 0.3420201;
    y[2][0] = 0.4226183;
    y[3][0] = 0.5000000;
    y[4][0] = 0.5735764;
    y[5][0] = 0.6427876;

    // Calculating the forward difference
    // table
    for (int i = 1; i < n; i++)
    {
        for (int j = 0; j < n - i; j++)
            y[j][i] = y[j + 1][i - 1] - y[j][i - 1];
    }

 cout<<setw(10)<<"x"<<setw(16)<<"sin(x)"<<endl;
    // Displaying the forward difference table
    for (int i = 0; i < n; i++) {
        cout << setw(10) << x[i]  //setw ussage
            << "\t";
        for (int j = 0; j < n - i; j++)
            cout << setw(10) << y[i][j]
                << "\t";
        cout << endl;
    }

    return 0;
}
```

Output

```
PS C:\Users\thegr\Desktop\DU\Sem 4> cd "c:\Users\thegr\Desktop\DU\Sem 4\Sec Lab\" ; if ($?) { g++ ForwardInte
ion }
         x          sin(x)
        15        0.258819       0.0832011      -0.00260288      -0.00061363      2.48253e-05      4.08292e-06
        20         0.34202       0.0805982      -0.00321651     -0.000588804      2.89083e-05
        25        0.422618       0.0773817      -0.00380531     -0.000559896
        30             0.5       0.0735764      -0.00436521
        35        0.573576       0.0692112
        40        0.642788
```

Code (BACKWARD DIFFERENCE)

```
// newton backward interpolation
#include <bits/stdc++.h>
using namespace std;


int main()
{
    // number of values given
    int n = 6;
    float x[] = { 15, 20, 25, 30, 35, 40 };

    // y[][] is used for difference
    // table and y[][0] used for input
    float y[n][n];
    y[0][0] = 0.2588190;
    y[1][0] = 0.3420201;
    y[2][0] = 0.4226183;
    y[3][0] = 0.5000000;
    y[4][0] = 0.5735764;
    y[5][0] = 0.6427876;

    // Calculating the backward difference table
    for (int i = 1; i < n; i++) {
        for (int j = n - 1; j >= i; j--)
            y[j][i] = y[j][i - 1] - y[j - 1][i - 1];
    }
cout<<setw(10)<<"x"<<setw(16)<<"sin(x)"<<endl;
    // Displaying the backward difference table
    for (int i = 0; i < n; i++) {
        cout << setw(10) << x[i]  //setw ussage
            << "\t";
        for (int j = 0; j <= i; j++)
            cout << setw(10) << y[i][j]
                << "\t";
        cout << endl;
    }

    return 0;
}
```

Output

```
PS C:\Users\thegr\Desktop\DU\Sem 4> cd "c:\Users\thegr\Desktop\DU\Sem 4\Sec Lab\" ; if ($?) { g++ BacwardInterp
ion }
        x          sin(x)
       15        0.258819
       20         0.34202       0.0832011
       25        0.422618       0.0805982      -0.00260288
       30             0.5       0.0773817      -0.00321651     -0.00061363
       35        0.573576       0.0735764      -0.00380531     -0.000588804    2.48253e-05
       40        0.642788       0.0692112      -0.00436521     -0.000559896    2.89083e-05     4.08292e-06
```