Question 1: *// Write a program to implement the Lagrange interpolation*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

double Li(int *i*, int *total\_given\_values*, double *x*[*total\_given\_values* + 1], double *find\_value*) *// Li(x) is the weighting function*

{

    double product = 1;

    for (int j = 0; j <= *total\_given\_values*; j++)

        if (j != *i*)

            product \*= (*find\_value* - *x*[j]) / (*x*[*i*] - *x*[j]);

    return product;

}

double Fn(int *total\_given\_values*, double *x*[*total\_given\_values* + 1], double *y*[*total\_given\_values* + 1], double *find\_value*) *// Fn(x) --> Lagrangian interpolating polynomial*

{

    double sum = 0;

    for (int i = 0; i <= *total\_given\_values*; i++)

        sum += (Li(i, *total\_given\_values*, *x*, *find\_value*) \* *y*[i]);

    return sum;

}

void main()

{

    int total\_given\_values;

    printf("Enter the total number of given values: ");

    scanf("*%d*", &total\_given\_values);

    total\_given\_values--; *// as we start counting from 0*

    double x[total\_given\_values + 1], y[total\_given\_values + 1], find\_value; *// arrays to store x and y(x) and find\_value is the value of x at which we have to find f(x)*

    for (int i = 0; i <= total\_given\_values; i++)

    {

        printf("Enter the value of x*%d*: ", i);

        scanf("*%lf*", &x[i]);

        printf("Enter the value of y*%d*: ", i);

        scanf("*%lf*", &y[i]);

        printf("\n");

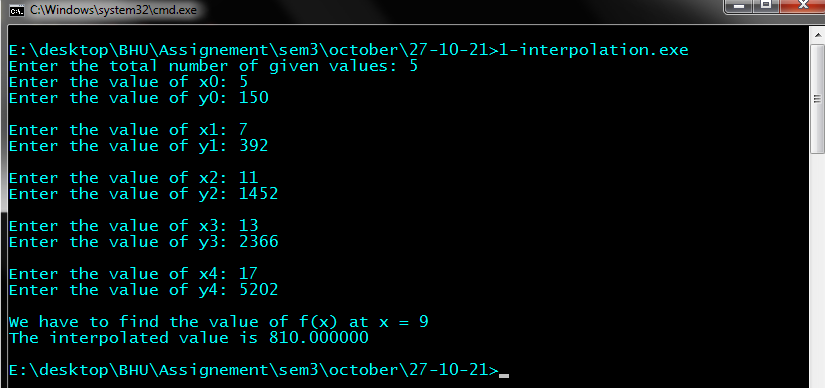
    }

    printf("We have to find the value of f(x) at x = ");

    scanf("*%lf*", &find\_value);

    printf("The interpolated value is *%lf*\n", Fn(total\_given\_values, x, y, find\_value));

}

Solution: 

Question 2: *// Write a program to implement Newton's Divided Difference table*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

void findDividedDifference(float *x*[], float *y*[][10], int *total\_given\_values*);

void printDifferenceTable(float *x*[], float *y*[][10], int *total\_given\_values*);

int main()

{

    int total\_given\_values;

    printf("Enter total given values: ");

    scanf("*%d*", &total\_given\_values);

    float y[10][10], x[total\_given\_values];

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("Enter the value of x*%d*: ", i);

        scanf("*%f*", &x[i]);

        printf("Enter the value of f(x*%d*): ", i);

        scanf("*%f*", &y[i][0]);

        printf("\n");

    }

    findDividedDifference(x, y, total\_given\_values);

    printDifferenceTable(x, y, total\_given\_values);

    return 0;

}

void findDividedDifference(float *x*[], float *y*[][10], int *total\_given\_values*)

{

    for (int i = 1; i < *total\_given\_values*; i++)

        for (int j = 0; j < *total\_given\_values* - i; j++)

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

}

void printDifferenceTable(float *x*[], float *y*[][10], int *total\_given\_values*)

{

    int del = 30;

printf("x\t\tf(x)\t\t%c\t\t%c2\t\t%c3\n", del, del, del);

    for (int i = 0; i < *total\_given\_values*; i++)

    {

        printf("*%f*\t", *x*[i]);

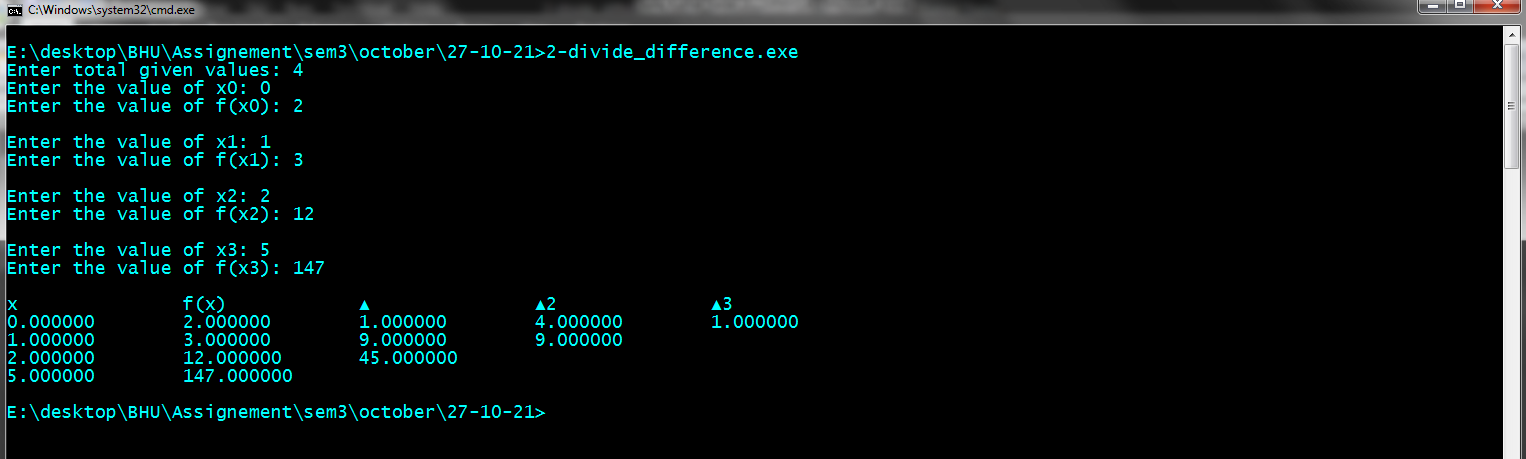
        for (int j = 0; j < *total\_given\_values* - i; j++)

            printf("*%f*\t", *y*[i][j]);

        printf("\n");

    }

}

Output: 

Question 3: *// Write a program to implement Forward Difference Table*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

int main()

{

    float function\_and\_values[100][101];

    int total\_given\_values, del = 30; *// character "del" will print triangle like structure*

    printf("\nEnter the total number of given values: ");

    scanf("*%d*", &total\_given\_values);

    printf("\n");

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("Enter the value of x*%d*: ", i);

        scanf("*%f*", &function\_and\_values[i][0]);

        printf("Enter the value of y*%d*: ", i);

        scanf("*%f*", &function\_and\_values[i][1]);

        printf("\n");

    }

*// Forming Forward difference table*

    for (int j = 2; j <= total\_given\_values; j++)

        for (int i = 0; i < total\_given\_values - 1; i++)

            function\_and\_values[i][j] = function\_and\_values[i + 1][j - 1] - function\_and\_values[i][j - 1];

*// Printing table*

    printf("\nNewton's Forward Difference table is:\n");

    printf("\nx\t\tf(x)\t\t");

    for (int i = 0; i <= total\_given\_values - 2; i++)

        printf("*%c*^*%d*\t\t", del, i + 1);

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("\n");

        for (int j = 0; j < total\_given\_values + 1 - i; j++)

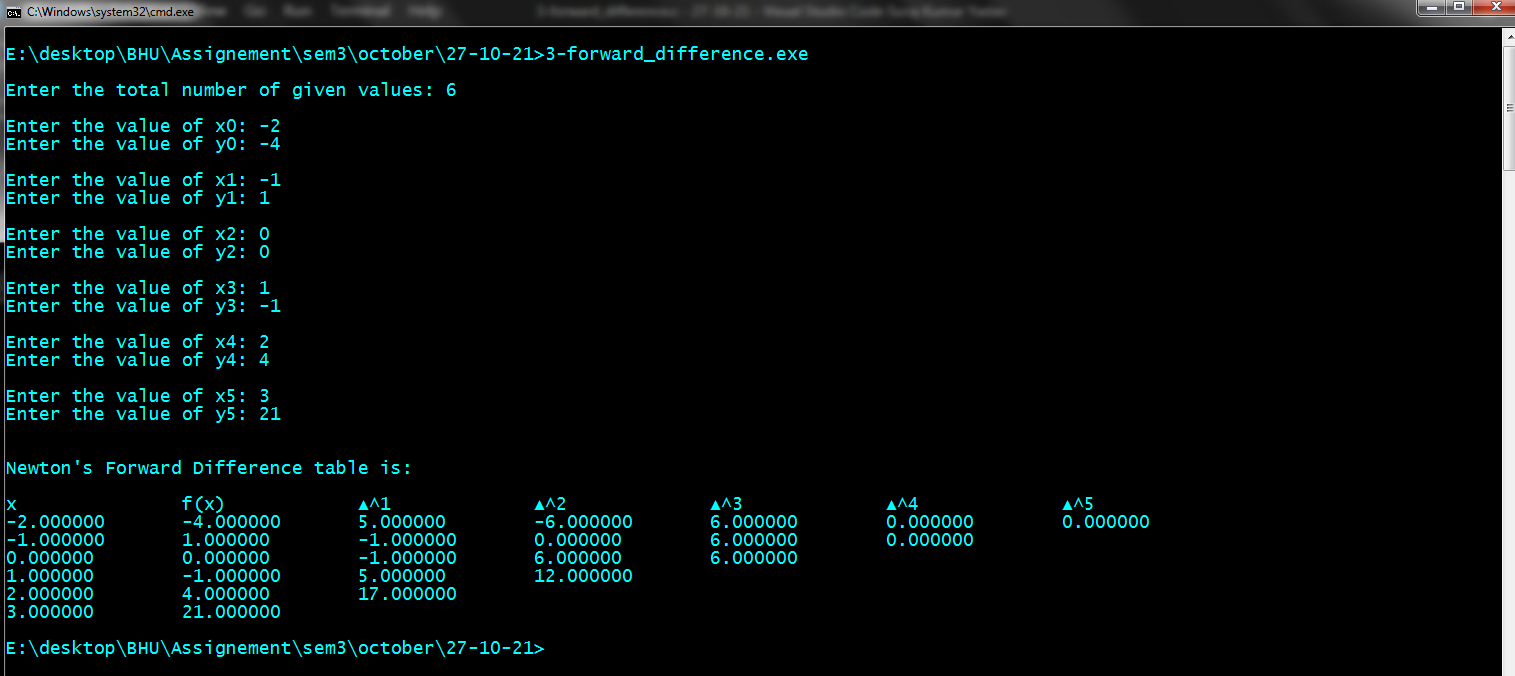
            printf("*%f*\t", function\_and\_values[i][j]);

    }

    printf("\n");

    return 0;

}

Output: 

Question 4: *// Write a program to implement Backward Difference Table*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

int main()

{

    float function\_and\_values[100][101];

    int total\_given\_values, del = 31;

    printf("\nEnter the total number of given values: ");

    scanf("*%d*", &total\_given\_values);

    printf("\n");

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("Enter the value of x*%d*: ", i);

        scanf("*%f*", &function\_and\_values[i][0]);

        printf("Enter the value of y*%d*: ", i);

        scanf("*%f*", &function\_and\_values[i][1]);

        printf("\n");

    }

*// Forming Backward difference table*

    for (int j = 2; j <= total\_given\_values; j++)

        for (int i = 0; i < total\_given\_values - 1; i++)

            function\_and\_values[i][j] = function\_and\_values[i + 1][j - 1] - function\_and\_values[i][j - 1];

*// Printing table*

    printf("\nNewton's Backward Difference table is:\n");

    printf("\nx\t\tf(x)\t\t");

    for (int i = 0; i <= total\_given\_values - 2; i++)

        printf("*%c%d*\t\t", del, i + 1);

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("\n");

        for (int j = 0; j < total\_given\_values + 1 - i; j++)

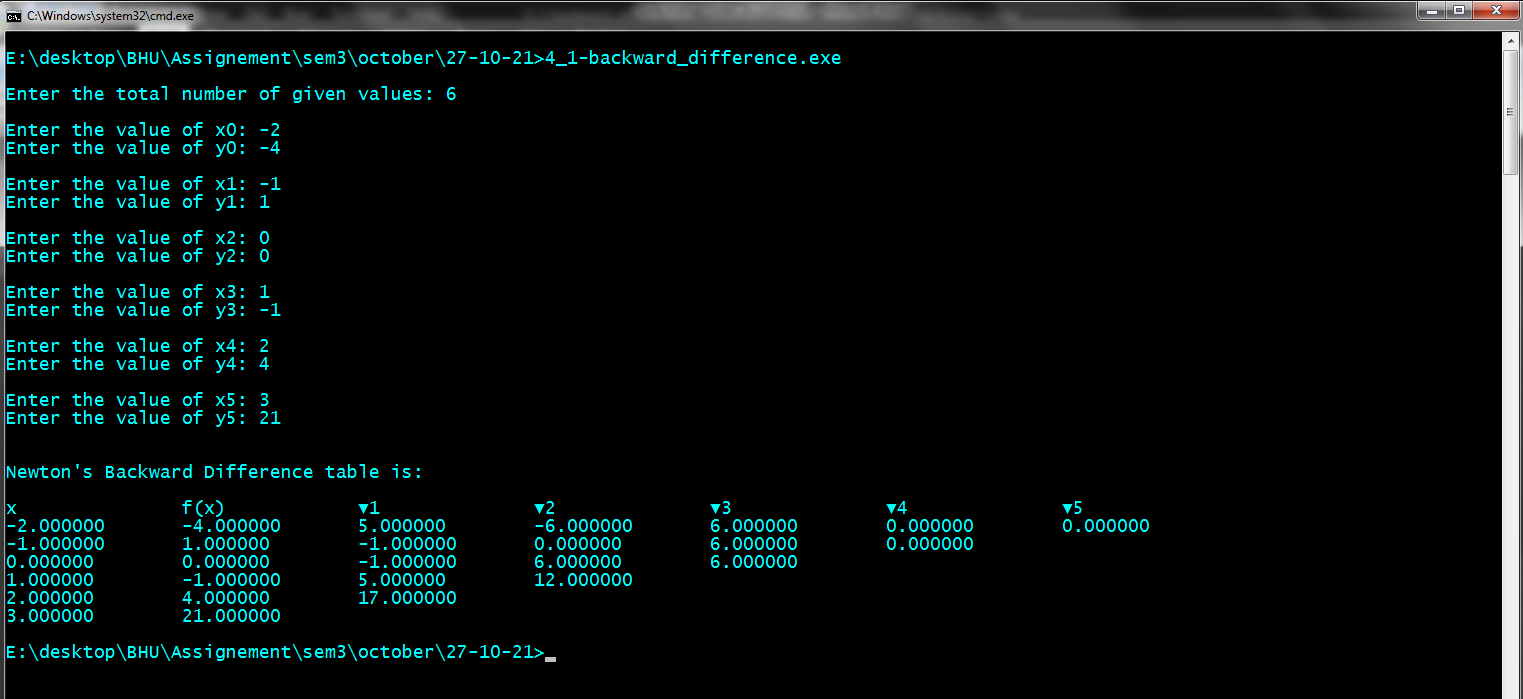
            printf("*%f*\t", function\_and\_values[i][j]);

    }

    printf("\n");

    return 0;

}

Output: 

Question 5: *// Write a program to calculate the value of function at some intermediate point using Newton's Backward Interpolation formula*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

int fact(int *total\_given\_values*)

{

    int f = 1, i;

    for (i = 1; i <= *total\_given\_values*; i++)

        f = f \* i;

    return f;

}

int main()

{

    int total\_given\_values, del = 31;

    float function\_and\_values[10][11], product\_of\_u\_and\_fx = 1, x, y, u, h; *// x for the value to find and y = f(x), h is interval for equally spaced data, u = (x-x0)/h, initial value of y = f(x0)*

    printf("\nEnter the total number of given values: ");

    scanf("*%d*", &total\_given\_values);

    printf("\n");

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("Enter the value of x*%d*: ", i);

        scanf("*%f*", &function\_and\_values[i][0]);

        printf("Enter the value of y*%d*: ", i);

        scanf("*%f*", &function\_and\_values[i][1]);

        printf("\n");

    }

*// Forming Backward difference table*

    for (int j = 2; j <= total\_given\_values; j++)

        for (int i = 0; i < total\_given\_values - 1; i++)

            function\_and\_values[i][j] = function\_and\_values[i + 1][j - 1] - function\_and\_values[i][j - 1];

*// Printing table*

    printf("\nNewton's Backward Difference table is:\n");

    printf("\nx\t\tf(x)\t\t");

    for (int i = 0; i <= total\_given\_values - 2; i++)

        printf("*%c*^*%d*\t\t", del, i + 1);

    for (int i = 0; i < total\_given\_values; i++)

    {

        printf("\n");

        for (int j = 0; j < total\_given\_values + 1 - i; j++)

            printf("*%f*\t", function\_and\_values[i][j]);

    }

    printf("\n\nEnter value of x to be calculated: ");

    scanf("*%f*", &x);

*// Calculating f(x)*

    h = function\_and\_values[total\_given\_values - 1][0] - function\_and\_values[total\_given\_values - 2][0];

    u = (x - function\_and\_values[total\_given\_values - 1][0]) / h;

    y = function\_and\_values[total\_given\_values - 1][1];

    for (int i = 1; i < total\_given\_values; i++)

    {

        product\_of\_u\_and\_fx \*= (u + (i - 1));

        y += (function\_and\_values[total\_given\_values - 1 - i][i + 1] \* product\_of\_u\_and\_fx) / fact(i);

    }

    printf("\nFinal Result is:\nf(*%f*) = *%f*\n", x, y);

    return 0;

}

Output: 