

[Date]

Lagrange Interpolation Polynomial

Lab Assignment: 02

Course Code: CSE 224

Course Title: Numerical Analysis Lab

Submitted to:

Name: Adeeba Anis

Lecturer

Department of CSE

at Bangladesh University of Business and Technology.

Submitted by:

Name: Syeda Nowshin Ibnat

ID: 17183103020

Intake: 39

Section: 1

Program: B.Sc. in CSE

Date of Submission: 13.07.2020

**Experiment no:** 7

**Name of Experiment:** Table below gives values of square of integers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 1 | 2 | 3 | 4 | 5 |
| sqr(x) | 1 | 4 | 9 | 16 | 25 |

Write a C/ C++ program to estimate the value of sqr(1.5) using Lagrange interpolation polynomial.

**Objective:** Use Lagrange Interpolation Polynomial to find the square of x

here is=1.5.

**Theory:** In [numerical analysis](https://en.wikipedia.org/wiki/Numerical_analysis), Lagrange polynomials are used for [polynomial interpolation](https://en.wikipedia.org/wiki/Polynomial_interpolation). For a given set of points (xi, f(x)) {\displaystyle (x\_{j},y\_{j})} with no two xi {\displaystyle x\_{j}}values equal, the Lagrange polynomial is the polynomial of lowest [degree](https://en.wikipedia.org/wiki/Degree_of_a_polynomial) that assumes at each value xi{\displaystyle x\_{j}} the corresponding value f(x){\displaystyle y\_{j}}, so that the functions coincide at each point.

The Lagrange form of the interpolation polynomial shows the linear character of polynomial interpolation and the uniqueness of the interpolation polynomial. Therefore, it is preferred in proofs and theoretical arguments. The Lagrange interpolating polynomial is the [polynomial](https://mathworld.wolfram.com/Polynomial.html) P(x) of degree <=(n-1) that passes through the n points (x_1,y_1=f(x_1)), (x_2,y_2=f(x_2)), ..., (x_n,y_n=f(x_n)), and is given by

|  |
| --- |
| P(x)=sum_(j=1)^nP_j(x), |

where

|  |
| --- |
| P_j(x)=y_jproduct_(k=1; k!=j)^n(x-x_k)/(x_j-x_k). |

Written explicitly,

|  |  |  |
| --- | --- | --- |
| P(x) | = | ((x-x_2)(x-x_3)...(x-x_n))/((x_1-x_2)(x_1-x_3)...(x_1-x_n))y_1+((x-x_1)(x-x_3)...(x-x_n))/((x_2-x_1)(x_2-x_3)...(x_2-x_n))y_2+...+((x-x_1)(x-x_2)...(x-x_(n-1)))/((x_n-x_1)(x_n-x_2)...(x_n-x_(n-1)))y_n. |

**Method:**

1. Start

2. Read number of data (n)

3. Read data Xi and Yi for i=1 to n

4. Read value of independent variables say xp

whose corresponding value of dependent say yp is to be determined.

5. Initialize: yp = 0

6. For i = 1 to n

Set p = 1

For j =1 to n

If i ≠ j then

Calculate p = p \* (xp - Xj)/(Xi - Xj)

End If

Next j

Calculate yp = yp + p \* Yi

Next i

6. Display value of yp as interpolated value.

7. Stop

**C Program:**

#include<stdio.h>

#include<math.h>

int main( {

float x[10],y[10],a,s=1,t=1,k=0;

int n,i,j,d=1;

printf("\nEnter the number of the terms of the table: ");

scanf("%d",&n);

printf("\n\nEnter the given values of x and y: \n");

for(i=0; i<n; i++) {

scanf ("%f",&x[i]);

scanf("%f",&y[i]); }

printf("\n\nThe table can write as follows :\n\n");

for(i=0; i<n; i++) {

printf("%0.2f\t%0.2f",x[i],y[i]);

printf("\n"); }

while(d==1) {

printf("\n\n\n Enter the value of the x to find the value of y\n\n\n");

scanf("%f",&a);

for(i=0; i<n; i++) {

s=1;

t=1;

for(j=0; j<n; j++) {

if(j!=i) {

s=s\*(a-x[j]);

t=t\*(x[i]-x[j]); } }

k=k+((s/t)\*y[i]); }

printf("\n\nThe value of y is: %0.2f",k); }

return 0; }

**Output:**

