Lagrange's Interpolation method:

Interpolation is the art of Reading between the line of a table or it means a process of computing intermediate Values of a function from the given set of tabulate Values of the function.

Suppose the table represent a set of Values of x and y

x : x_0 x_1 x_2 ... x_n

y = y_0 y_1 y_2 ... y_n

Now if we require a Value $y = y_i$ corresponding to a value $x = x_i$. where $x_0 < x_i < x_n$.

Extrapolation: The process of finding the Values outside the interval (x_0, x_n). But, in General we use interpolation.

$y = f(x)$

In Lagrange's method independent variable are not equally spaced and in cases when the differences of the dependent are not small, we use Lagrange's interpolation formula.

Let $y = f(x)$

where $f(x)$ can take the values $y_0, y_1, y_2 \dots y_n$ corresponding to $x = x_0, x_1, x_2 \dots x_n$. i.e., $y_i = f(x_i)$ $i = 0, 1, 2 \dots n$.

$(n+1)$ paired values of (x_i, y_i), $i = 0, 1, 2, \dots n$.

& $f(x)$ can be represented by a polynomial function of degree n.

$$y = f(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} \times y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} \times y_1$$

$$+ \dots + \frac{(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} y_n.$$

— unequal intervals of the independent variable.

Algorithm :

1. Start
2. Read number of data (n).
3. Read x_i and y_i for $i = 1$ to n.
4. Read the value of the independent variables say x_p whose corresponding value of dependent say y_p to be determined.
5. Initialize $y_p = 0$
6. For $i = 1$ to n

   Set $p = 1$

   For $j = 1$ to n

   if $i \neq j$ then

   calculate $p = p * (x_p - x_j) / (x_i - x_j)$

   end if

   next j

   Calculate $y_p = y_p + p * y_i$
7. Display the value of y_p as interpolated value.
8. Stop.

Pseudocode:

1. Start
2. Read the number of data (n).
3. Read data :

    for $i = 1$ to n

      Read x_i and y_i

    next i

4. Read x_p
5. Initialize $y_p = 0$
6. For $i = 1$ to n

    $p = 1$

    For $j = 1$ to n

      if ($i \neq j$)

      $p = p * (x_p - x_j) / (x_i - x_j)$

      endif

next j

$y_p = y_p + p * y_i$

next i

7) print y_p

8) Stop

Lagrange's Interpolation Program:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
Void main()
{  float  x[100],y[100], xp=0; yp=0, p;
   int i, j, n;

  /* Input Section */
```

```
printf ("Enter the number of data\n");
scanf ("%d", &n);
printf ("Enter the data \n");
for (i=1, i<=n; i++)
{
    printf ("x[%d]", i);
    scanf ("%f", &x[i]);
    printf ("y[%d]", i);
    scanf ("%f", &y[i]);
}
printf ("Enter the interpolation point\n");
scanf ("%f", &xp);
```

```
/* implementing Lagrange's interpolation */
for (i = 0; i <= n; i++)
{
    p = 1;
    for (j = 1; j <= n; j++)
    {
        if (i != j)
        {
            p = p * (xp - x[j]) / (x[i] - x[j]);
        }
    }
    yp = yp + p * y[i];
}
printf("Interpolated Value at %0.3f is %0.3f", xp, yp);
```

```
getch ( ) ;
}.
```

1. Using C program for Lagrange's interpolation to find y[10] from the following table.

x:	5	6	9	11
y:	12	13	14	16

y(10) is (14.6666)

2. Using C-program for Lagrange's interpolation to find y(9.5) given that

x:	7	8	9	10
y:	3	1	1	9

y(9.5) = 3.625.

3. The following are the measurements t made on a curve recorded by the oscillograph representing

a change of current i due to the change in the condition of the electrical circuits.

t	1.2	2.0	2.5	3
i	1.36	0.58	0.34	0.20

Using C program for Lagrange's interpolation to find i at $t = 1.6$ $\boxed{i = 0.8932}$