Practical – 1

Calculate the SUM :

```
Sum[1 / i, {i, 1, n}]
HarmonicNumber[n]
Table[HarmonicNumber[n], {n, 1, 20}]
```

$$\frac{137}{60}$$

$$\frac{137}{60}$$

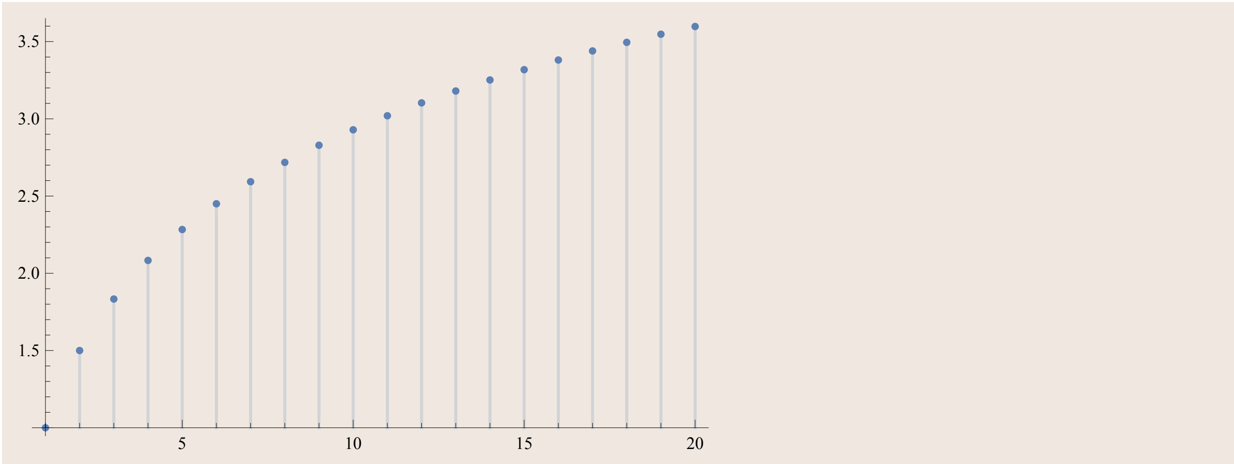
$$\left\{1, \frac{3}{2}, \frac{11}{6}, \frac{25}{12}, \frac{137}{60}, \frac{49}{20}, \frac{363}{140}, \frac{761}{280}, \frac{7129}{2520}, \frac{7381}{2520}, \frac{83\,711}{27\,720}, \frac{86\,021}{27\,720}, \frac{1\,145\,993}{360\,360}, \frac{1\,171\,733}{360\,360}, \frac{1\,195\,757}{360\,360}, \frac{2\,436\,559}{720\,720}, \frac{42\,142\,223}{12\,252\,240}, \frac{14\,274\,301}{4\,084\,080}, \frac{275\,295\,799}{77\,597\,520}, \frac{55\,835\,135}{15\,519\,504}\right\}$$

$$\sum_{n=1}^{\infty} \text{HarmonicNumber}[n]$$

Sum::div : Sum does not converge. ≫

$$\sum_{n=1}^{\infty} \text{HarmonicNumber}[n]$$

```
DiscretePlot[HarmonicNumber[n], {n, 1, 20}]
```




```
num = Input["Give an integer:"]; sum = 0; For[i = 1, i ≤ num, i++, sum += (1 / i);
 Print["The harmonic sum of first", i, "numbers is:", sum]]
```

The harmonic sum of first1numbers is:1

The harmonic sum of first2numbers is: $\frac{3}{2}$

The harmonic sum of first3numbers is: $\frac{11}{6}$

The harmonic sum of first4numbers is: $\frac{25}{12}$

The harmonic sum of first5numbers is: $\frac{137}{60}$

The harmonic sum of first6numbers is: $\frac{49}{20}$

The harmonic sum of first7numbers is: $\frac{363}{140}$

The harmonic sum of first8numbers is: $\frac{761}{280}$

The harmonic sum of first9numbers is: $\frac{7129}{2520}$

The harmonic sum of first10numbers is: $\frac{7381}{2520}$

Practical – 2

To Find the absolute value of integer :

Type – 1

```
num = Input["Enter the number"]
If[num ≥ 0, Print["The abslute value of", num, "is: ", num],
 Print["The absolute value of ", num, "is: ", -num]]
```

```
-3
```

```
The absolute value of -3is: 3
```

Type – 2

```
num1 = Input["Give an integer:"]; Print["The number you have suppled is ", num1];
If[num1 > 0, Print["The absolute value of ", num, "is:", num],
 Print["The absolute value of ", num1, "is: ", -num1]]
```

```
The number you have suppled is -100
The absolute value of -100is: 100
```

Type – 3

```
abs[x_] := If[x > 0, Print["The absolute value of ", x, " is :", x],
  Print["The absolute value of ", x, " is :", -x]]
abs[
 -Pi]
```

```
The absolute value of -π is :π
```

```
N[Abs[-Pi], 16]
```

```
3.141592653589793
```

Practical-3

Sort A List :

```
Sort[{d, b, c, a}]
```

```
{a, b, c, d}
```

Sort using Greater as the ordering function :

```
Sort[{4, 1, 3, 2}, Greater]
```

```
{4, 3, 2, 1}
```

```
Sort[{4, 1, 3, 2}, Less]
```

```
{1, 2, 3, 4}
```

```
Sort[{4, 1, 3, 2}, #1 > #2 &]
```

```
{4, 3, 2, 1}
```

```
Sort[{4, 1, 3, 2}, #1 < #2 &]
```

```
{1, 2, 3, 4}
```

Sort by comparing he second part of each element :

```
Sort[{{a, 2}, {c, 1}, {d, 3}}, #1[[2]] < #2[[2]] &]
```

```
{{c, 1}, {a, 2}, {d, 3}}
```

```
Sort[{"cat", "fish", "catfish", "Cat"}]
```

```
{cat, Cat, catfish, fish}
```

Sort by numerical value :

```
Sort[{Pi, E, 2, 3, 1, Sqrt[2]}, Less]
```

$\left\{1, \sqrt{2}, 2, e, 3, \pi\right\}$

Sort by absolute value :

```
Sort[{-11, 10, 2, 1, -4}, Abs[#1] < Abs[#2] &]
```

```
{1, 2, -4, 10, -11}
```

```
RandomInteger[{-5, 5}, {20, 1}]
```

```
{{2}, {5}, {0}, {-2}, {0}, {0}, {3}, {-4}, {2}, {4},
 {-2}, {5}, {5}, {3}, {3}, {-4}, {0}, {3}, {-5}, {5}}
```

```
Sort[%]
```

```
{{-5}, {-4}, {-4}, {-2}, {-2}, {0}, {0}, {0}, {0},
 {2}, {2}, {3}, {3}, {3}, {3}, {4}, {5}, {5}, {5}, {5}}
```

```
RandomInteger[{-5, 5}, {10, 3}]
```

```
{{-4, 3, 0}, {1, -4, 0}, {-4, 3, -5},
 {1, -5, 4}, {4, -4, -2}, {5, -3, 2}, {4, 4, 1},
 {-4, -4, 3}, {4, -2, -3}, {3, -5, 0}}
```

```
Sort[%, Norm[#1] < Norm[#2] &]
```

```
{{1, -4, 0}, {-4, 3, 0}, {4, -2, -3},
 {4, 4, 1}, {3, -5, 0}, {4, -4, -2}, {5, -3, 2},
 {-4, -4, 3}, {1, -5, 4}, {-4, 3, -5}}
```

```
Sort[{I, 1 + I, 1 - I, 2 + 3 I}, Re[#1] < Re[#2] &]
```

```
{ⅈ, 1 - ⅈ, 1 + ⅈ, 2 + 3 ⅈ}
```

```
Sort[{4, 3, 2, 1}, (Print[{#1, #2}]; #1 > #2) &]
```

```
{4, 3}
{2, 1}
{3, 2}
```

```
{4, 3, 2, 1}
```

```
list = RandomInteger[10, 10]
```

```
{0, 7, 6, 1, 3, 9, 7, 9, 9, 8}
```

```
Sort[list]
```

```
{0, 1, 3, 6, 7, 7, 8, 9, 9, 9}
```

```
list[[Ordering[list]]]
```

```
{0, 1, 3, 6, 7, 7, 8, 9, 9, 9}
```

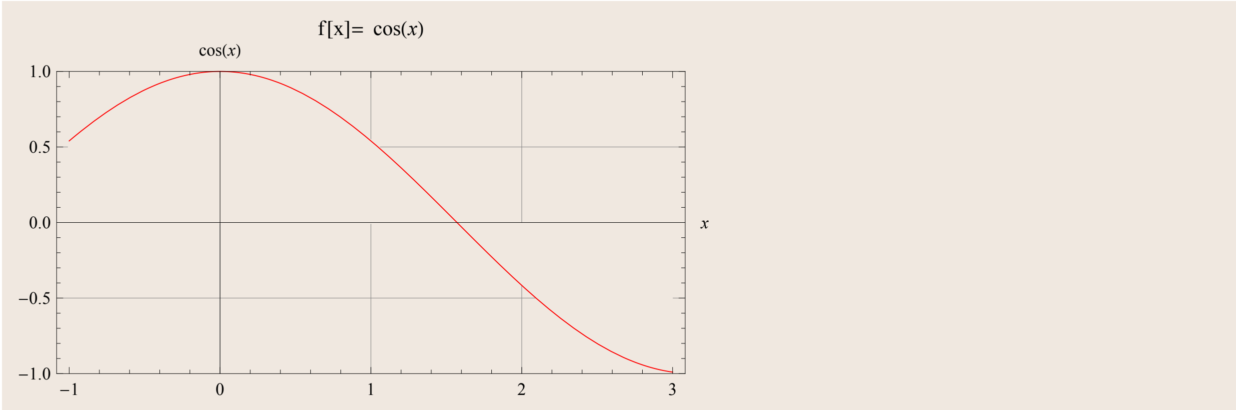
Practical – 4

Bisection Method :

```
x0 = 0;
x1 = 2.0;
Nmax = 20;
eps = 0.0001;
f[x_] := Cos[x];
If[N[f[x0] * f[x1]] > 0,
 Print["These values do not satisfy the IVP so change the values."],
     For [i = 1, i <= Nmax, i++,
        a = (x0 + x1) / 2;
  If [Abs[(x1 - x0) / 2] < eps, Return[a],   Print[i, "th iteration value is : ", a];
   Print[" Estimated error in ", i, "th iteration is : ", (x1 - x0) / 2];
   If [f[a] * f[x1] > 0, x1 = a , x0 = a]]];
Print[" Root is : ", a];
Print[" Estimated error in ", i, "th iteration is : ", (x1 - x0) / 2]]
Plot[f[x], {x, -1, 3}, PlotRange → {-1, 1},
 PlotStyle → Red, PlotLabel → "f[x]= " f[x], AxesLabel → {x, f[x]},
 AspectRatio → Automatic, Frame → True, GridLines → Automatic,
 ClippingStyle → Automatic, Filling → Axis, FillingStyle → LightBrown]
```

```
1th iteration value is : 1.

 Estimated error in 1th iteration is : 1.

2th iteration value is : 1.5

 Estimated error in 2th iteration is : 0.5

3th iteration value is : 1.75

 Estimated error in 3th iteration is : 0.25

4th iteration value is : 1.625

 Estimated error in 4th iteration is : 0.125

5th iteration value is : 1.5625

 Estimated error in 5th iteration is : 0.0625

6th iteration value is : 1.59375

 Estimated error in 6th iteration is : 0.03125

7th iteration value is : 1.57813

 Estimated error in 7th iteration is : 0.015625

8th iteration value is : 1.57031

 Estimated error in 8th iteration is : 0.0078125

9th iteration value is : 1.57422

 Estimated error in 9th iteration is : 0.00390625

10th iteration value is : 1.57227

 Estimated error in 10th iteration is : 0.00195313

11th iteration value is : 1.57129

 Estimated error in 11th iteration is : 0.000976563

12th iteration value is : 1.5708

 Estimated error in 12th iteration is : 0.000488281

13th iteration value is : 1.57056

 Estimated error in 13th iteration is : 0.000244141

14th iteration value is : 1.57068

 Estimated error in 14th iteration is : 0.00012207
```

```
Return[1.57074]
```



the value of a is 1.

the value of a is 2.

the value of a is 0.001.

The value of Nmax is 10.

Iterations no. Approximation

1 1.5

2 1.5

3 1.5

4 1.5

5 1.5

6 1.5

7 1.5

8 1.5

9 1.5

10 1.5

The maximum number of iterations has been achieved prior to given error tolerance

The approximate value of te root is 1.5 and f[m]=0.0707372 − 1.5 $\mathrm{Exp}^{3/2}$

BISECTION METHOD : BY CONSTRUCTING THE FUNCTION

```
bisection[f_, x0_, x1_, Nmax_, eps_] := If[N[f[x0] * f[x1], 16] > 0,
  Print["These values do not satisfy the IVP so change the values."],
  y0 = x0; y1 = x1; For[ i = 1, i <= Nmax , i ++,
   a = (y0 + y1) / 2; If [Abs[(y1 - y0) / 2] < eps, Return[N[a, 8]],
    Print[i, "th iteration value is : ", N[a, 16]]; Print[" Estimated error is : ",
     N[(y1 - y0) / 2, 8]]; If [f[a] * f[y1] > 0,  y1 = a,  y0 = a]]];
  Print[" Root is : ", N[a, 8]]; Print[" Estimated error is : ",
   N[(y1 - y0) / 2, 8]];
  Plot[f[x], {x, -1, 3}]]
f[x] := Cos[x]
bisection[f, 1, 2, 20, 0.0001]
```

```
1th iteration value is : 1.500000000000000
 Estimated error is : 0.50000000
2th iteration value is : 1.750000000000000
 Estimated error is : 0.25000000
3th iteration value is : 1.625000000000000
 Estimated error is : 0.12500000
4th iteration value is : 1.562500000000000
 Estimated error is : 0.062500000
5th iteration value is : 1.593750000000000
 Estimated error is : 0.031250000
6th iteration value is : 1.578125000000000
 Estimated error is : 0.015625000
7th iteration value is : 1.570312500000000
 Estimated error is : 0.0078125000
8th iteration value is : 1.574218750000000
 Estimated error is : 0.0039062500
9th iteration value is : 1.572265625000000
 Estimated error is : 0.0019531250
10th iteration value is : 1.571289062500000
 Estimated error is : 0.00097656250
11th iteration value is : 1.570800781250000
 Estimated error is : 0.00048828125
12th iteration value is : 1.570556640625000
 Estimated error is : 0.00024414063
13th iteration value is : 1.570678710937500
 Estimated error is : 0.00012207031
```

```
1.5707397
```

BISECTION METHOD :

FOR PARAMETERS TAKEN FROM A USER

```
x0 = Input["Enter first guess : "];
x1 = Input["Enter second guess : "];
Nmax = Input["Enter maximum number of iterations : "];
eps = Input["Enter a value of convergence parameter : "]; Print["x0=", x0];
Print["x1=", x1];
Print["Nmax=", Nmax];
Print["epsilon=", eps];
f[x] := Cos[x];
Print["f(x) := ", f[x]]; If[N[f[x0] * f[x1]] > 0,
 Print["These values do not satisfy the IVP so change the values."],
 For[i = 1, i <= Nmax, i++, a = (x0 + x1) / 2;
  If [Abs[(x1 - x0) / 2] < eps, Return[N[a, 16]], If[f[a] * f[x1] > 0, x1 = a, x0 = a]];
  Print[i, "th iteration value is : ", N[a, 16]];
  Print[" Estimated error is : ", N[x1 - x0, 16]]]; Print[" Root is : ", N[a, 16]];
 Print[" Estimated error is : ", N[x1 - x0, 16]]]; Plot[f[x], {x, -1, 3}]
```

```
x0=0

x1=2

Nmax=20

epsilon=0.00001

f(x) := Cos[x]

1th iteration value is : 1.000000000000000

 Estimated error is : 1.000000000000000

2th iteration value is : 1.500000000000000

 Estimated error is : 0.5000000000000000

3th iteration value is : 1.750000000000000

 Estimated error is : 0.2500000000000000

4th iteration value is : 1.625000000000000

 Estimated error is : 0.1250000000000000

5th iteration value is : 1.562500000000000

 Estimated error is : 0.06250000000000000

6th iteration value is : 1.593750000000000

 Estimated error is : 0.03125000000000000

7th iteration value is : 1.578125000000000

 Estimated error is : 0.01562500000000000

8th iteration value is : 1.570312500000000
```

```
 Estimated error is : 0.00781250000000000
9th iteration value is : 1.574218750000000
 Estimated error is : 0.00390625000000000
10th iteration value is : 1.572265625000000
 Estimated error is : 0.00195312500000000
11th iteration value is : 1.571289062500000
 Estimated error is : 0.00097656250000000
12th iteration value is : 1.570800781250000
 Estimated error is : 0.000488281250000000
13th iteration value is : 1.570556640625000
 Estimated error is : 0.000244140625000000
14th iteration value is : 1.570678710937500
 Estimated error is : 0.000122070312500000
15th iteration value is : 1.570739746093750
 Estimated error is : 0.0000610351562500000
16th iteration value is : 1.570770263671875
 Estimated error is : 0.0000305175781250000
17th iteration value is : 1.570785522460938
 Estimated error is : 0.0000152587890625000
```

```
Return[1.570793151855469]
```

Practical -5

Newton's Method :

NEWTON METHOD : BY TAKING PARAMETERS AS INPUT

```
x0 = Input["Enter initial guess : "];
Nmax = Input["Enter maximum number of iterations : "];
eps = Input["Enter a value of convergence parameter : "];
Print["x0=", x0];
Print["Nmax=", Nmax];
Print["epsilon=", eps];
f[x] := Cos[x];
Print["f(x) := ", f[x]];
Print["f'(x):= ", f'[x]];
For[i = 1, i <= Nmax, i++, x1 = N[x0 - (f[x] /. x → x0) / (f'[x] /. x → x0)];
 If [Abs[x1 - x0] < eps, Return[x1], x0p = x0; x0 = x1];
 Print["In ", i, "th Number of iterations the approximation to root is : ", x1];
 Print[" Estimated error is : ", Abs[x1 - x0p]]];
Print["the final approximation of root is : ", x1];
Print[" Estimated error is : ", Abs[x1 - x0]]; Plot[f[x], {x, -1, 3}]
```

```
x0=1

Nmax=5

epsilon=0.00001

f(x) := Cos[x]

f'(x):= -Sin[x]

In 1th Number of iterations the approximation to root is : 1.64209

 Estimated error is : 0.642093

In 2th Number of iterations the approximation to root is : 1.57068

 Estimated error is : 0.0714173

In 3th Number of iterations the approximation to root is : 1.5708

 Estimated error is : 0.00012105
```

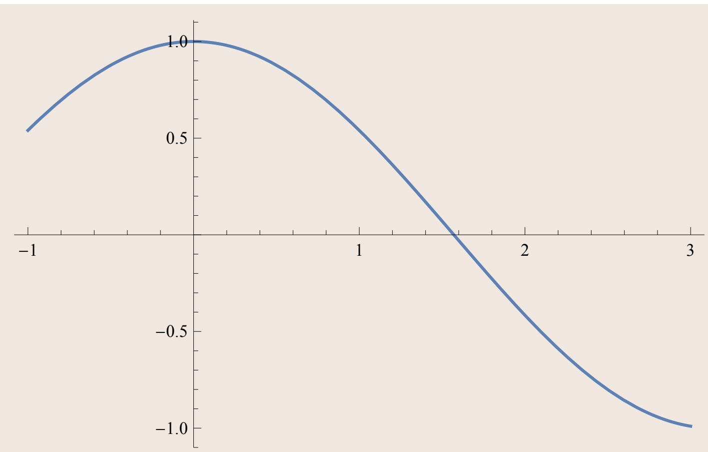
```
Return[1.5708]
```

NEWTON METHOD : BY GIVING VALUES OF PARAMETERS IN PROGRAM

```
x0 = 1;
Nmax = 5;
eps = 0.0001;
f[x] := Cos[x];
For[ i = 1, i <= Nmax, i++,
x1 = N[x0 - (f[x] /. x → x0) / (f'[x] /. x → x0)];
 If [Abs[x1 - x0] < eps, Return[x1], x0p = x0; x0 = x1];
 Print["In ", i, "th Number of iterations the approximation to root is : ", x1];
 Print[" Estimated error is : ", Abs[x1 - x0p]]];
Print["the final approximation of root is : ", x1];
Print[" Estimated error is : ", Abs[x1 - x0]];
Plot[f[x], {x, -1, 3}]
```

```
In 1th Number of iterations the approximation to root is : 1.64209

 Estimated error is : 0.642093

In 2th Number of iterations the approximation to root is : 1.57068

 Estimated error is : 0.0714173

In 3th Number of iterations the approximation to root is : 1.5708

 Estimated error is : 0.00012105
```

```
Return[1.5708]
```




```
newton[f_, x0_, Nmax_, eps_] :=
 Module[{y0, x, x1, x0p}, y0 = x0; Print["f(x)  := ", f[x]];
  Print["f'(x):= ", f'[x]]; For[ i = 1, i <= Nmax, i++, x1 = N[x0 - f[x0] / f'[x0]];
   If[Abs[x1 - x0] < eps, Return[x1], x0p = x0; x0 = x1]; Print["In ", i,
    "th Number of iterations the approximation to root is : ", x1];
   Print[" Estimated error is : ", Abs[x1 - x0p]]];
  Print["the final approximation of root is : ", x1];
  Print[" Estimated error is : ", Abs[x1 - x0]]; Plot[f[x], {x, -1, 3}]]
f[x] := Cos[x];
newton[f, 1, 5, 0.000001]
```

```
f(x) := f[x$2156]

f'(x):= f'[x$2156]
```

In 1th Number of iterations the approximation to root is : $1. - \frac{1.\,f[1.]}{f'[1.]}$

Estimated error is : $\text{Abs}\left[1. - \text{x0p\$2156} - \frac{1.\,f[1.]}{f'[1.]}\right]$

In 2th Number of iterations the approximation to root is : $1. - \frac{1.\,f[1.]}{f'[1.]}$

Estimated error is : $\text{Abs}\left[1. - \text{x0p\$2156} - \frac{1.\,f[1.]}{f'[1.]}\right]$

In 3th Number of iterations the approximation to root is : $1. - \frac{1.\,f[1.]}{f'[1.]}$

Estimated error is : $\text{Abs}\left[1. - \text{x0p\$2156} - \frac{1.\,f[1.]}{f'[1.]}\right]$

In 4th Number of iterations the approximation to root is : $1. - \frac{1.\,f[1.]}{f'[1.]}$

Estimated error is : $\text{Abs}\left[1. - \text{x0p\$2156} - \frac{1.\,f[1.]}{f'[1.]}\right]$

In 5th Number of iterations the approximation to root is : $1. - \frac{1.\,f[1.]}{f'[1.]}$

Estimated error is : $\text{Abs}\left[1. - \text{x0p\$2156} - \frac{1.\,f[1.]}{f'[1.]}\right]$

the final approximation of root is : $1. - \frac{1.\,f[1.]}{f'[1.]}$

Estimated error is : $\text{Abs}\left[0. - \frac{1.\,f[1.]}{f'[1.]}\right]$

Practical - 6

Regular Falsi Method :

METHOD OF FALSE POSITION : TAKING PARAMETERS AS INPUTS

```
x0 = Input["Enter first guess : "];
x1 = Input["Enter second guess : "];
Nmax = Input["Enter maximum number of iterations : "];
eps = Input["Enter a value of convergence parameter : "]; Print["x0=", x0];
Print["x1=", x1];
Print["Nmax=", Nmax];
Print["epsilon=", eps];
f[x] := Cos[x];
Print["f(x) := ", f[x]]; If[N[f[x0] * f[x1]] > 0,
 Print["These values do not satisfy the IVP so change the values."],
 For[i = 1, i <= Nmax, i++, a = N[x1 - f[x1] * (x1 - x0) / (f[x1] - f[x0]), 16];
  If [Abs[(x1 - x0) / 2] < eps, Return[N[a, 16]], Print[i, "th iteration value is : ",
    N[a, 16]]; Print[" Estimated error is : ", N[x1 - x0, 16]];
   If[f[a] * f[x1] > 0, x1 = a, x0 = a]]]; Print[" Root is : ", N[a, 16]];
 Print[" Estimated error is : ", N[x1 - x0, 16]]]; Plot[f[x], {x, -1, 3}]
```

```
x0=0

x1=2

Nmax=20

epsilon=0.00001

f(x) := Cos[x]

1th iteration value is : 1.412282927437392

 Estimated error is : 2.000000000000000

2th iteration value is : 1.573906323722879

 Estimated error is : 0.587717072562608

3th iteration value is : 1.570783521943903

 Estimated error is : 0.161623396285487

4th iteration value is : 1.57079632681545

 Estimated error is : 0.003122801778976
```

```
Return[1.57079632679490]
```

METHOD OF FALSE POSITION : CONSTRUCTING A FUNCTION

```
falseposition[f_, x0_, x1_, Nmax_, eps_] := If[N[f[x0] * f[x1]] > 0,
  Print["These values do not satisfy the IVP so change the values."],
  y0 = x0; y1 = x1; a = y1; If [Abs[y1 - y0] < eps, Return[N[a, 8]],
   For[ i = 1, i <= Nmax , i++, a = N[y1 - f[y1] * (y1 - y0) / (f[y1] - f[y0])];
    If [f[a] * f[y1] > 0, y1 = a, y0 = a]; If [Abs[y1 - y0] < eps, Return[N[a, 8]]];
    Print[i, "th iteration value is : ", N[a, 8]];
    Print[" Estimated error is : ", N[y1 - y0, 8]]]; Print[" Root is : ", N[a, 8]];
   Print[" Estimated error is : ", N[y1 - y0, 8]]]; Plot[f[x], {x, -1, 3}]]
f[x] := Cos[x];
falseposition[f, 0, 2, 5, 0.00000001]
```

```
1th iteration value is : 1.41228

 Estimated error is : 0.587717

2th iteration value is : 1.57391

 Estimated error is : 0.161623

3th iteration value is : 1.57078

 Estimated error is : 0.0031228

4th iteration value is : 1.5708

 Estimated error is : 0.0000128049
```

```
1.5708
```

METHOD OF FALSE POSITION : TAKING PARAMETERS

```
x0 = 0;
x1 = 2.0;
Nmax = 10;
eps = 0.000000000001;
f[x_] := Cos[x];
If[N[f[x0] * f[x1]] > 0,
  Print["These values do not satisfy the IVP so change the values."],
   If [Abs[x1 - x0] < eps, Return[N[x1, 8]],
   For[ i = 1, i <= Nmax , i++, a = N[x1 - f[x1] * (x1 - x0) / (f[x1] - f[x0]), 8];
    If [f[a] * f[x1] > 0, x1 = a, x0 = a]; If [Abs[x1 - x0] < eps, Return[N[a, 8]]];
    Print[i, "th iteration value is : ", N[a, 8]];
    Print[" Estimated error is : ", N[x1 - x0, 8]]]];
  Print[" Root is : ", N[a, 8]]; Print[" Estimated error is : ", N[x1 - x0, 8]]];
Plot[f[x], {x, -1, 3}, PlotRange → {-1, 1}, PlotStyle → Red, PlotLabel → "f[x]= " f[x],
 AxesLabel → {x, f[x]}, AspectRatio → Automatic, Frame → True, GridLines → Automatic,
 ClippingStyle → Automatic, Filling → Axis, FillingStyle → LightBrown]
```

1th iteration value is : 1.41228

Estimated error is : 0.587717

2th iteration value is : 1.57391

Estimated error is : 0.161623

3th iteration value is : 1.57078

Estimated error is : 0.0031228

4th iteration value is : 1.5708

Estimated error is : 0.0000128049

5th iteration value is : 1.5708

Estimated error is : 2.05567×10^{-11}

6th iteration value is : 1.5708

Estimated error is : 2.05567×10^{-11}

7th iteration value is : 1.5708

Estimated error is : 2.05567×10^{-11}

8th iteration value is : 1.5708

Estimated error is : 2.05567×10^{-11}

9th iteration value is : 1.5708

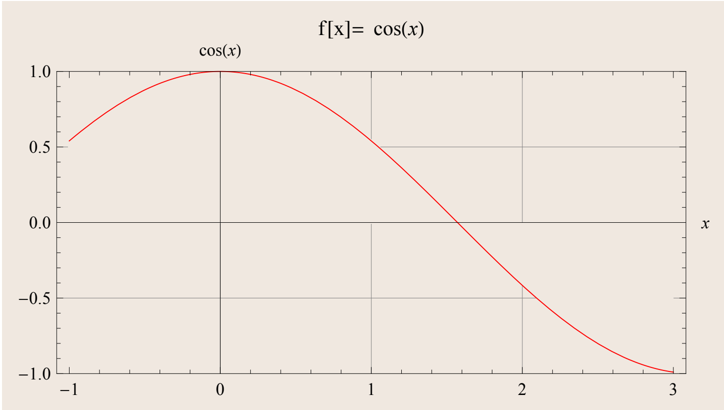
Estimated error is : 2.05567×10^{-11}

10th iteration value is : 1.5708

Estimated error is : 2.05567×10^{-11}

Root is : 1.5708

Estimated error is : 2.05567×10^{-11}



Practical - 7

LU Decomposition :

Question – 1 :

[redacted]

Solve the system of linear equation by using LU decomposition Method .

```
ClearAll[a, b, p, x, lu, c, l, u, d, luf]
a = {{1, 4, 3}, {2, 7, 9}, {5, 8, -2}};
MatrixForm[a]
b = {-4, -10, 9}
{lu, p, c} = LUDecomposition[a]
l = lu SparseArray[{i_, j_} /; j < i → 1, {3, 3}] + IdentityMatrix[3];
MatrixForm[l]
l = lu SparseArray[{i_, j_} /; j ≥ i → 1, {3, 3}]
MatrixForm[u]
d = MatrixForm[l.u]
luf = LinearSolve[a]
x = luf[b]
a.x
```

$$\begin{pmatrix} 1 & 4 & 3 \\ 2 & 7 & 9 \\ 5 & 8 & -2 \end{pmatrix}$$

```
{-4, -10, 9}
```

```
{{{1, 4, 3}, {2, -1, 3}, {5, 12, -53}}, {1, 2, 3}, 1}
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 5 & 12 & 1 \end{pmatrix}$$

```
{{1, 4, 3}, {0, -1, 3}, {0, 0, -53}}
```

```
u
```

```
{{1, 4, 3}, {0, -1, 3}, {0, 0, -53}}.u
```

```
LinearSolveFunction[{3, 3}, <>]
```

```
{3, -1, -1}
```

```
{-4, -10, 9}
```

Question – 2 :

Solve the system of linear equation by using LU decomposition Method .

```
ClearAll[a, b, p, x, lu, c, l, u, d, luf]
a = {{2, 7, 5}, {6, 20, 10}, {4, 3, 0}};
MatrixForm[a]
b = {0, 4, 1}
{lu, p, c} = LUDecomposition[a]
l = lu SparseArray[{i_, j_} /; j < i → 1, {3, 3}] + IdentityMatrix[3];
MatrixForm[l]
l = lu SparseArray[{i_, j_} /; j ≥ i → 1, {3, 3}]
MatrixForm[u]
d = MatrixForm[l.u]
luf = LinearSolve[a]
x = luf[b]
a.x
```

$$\begin{pmatrix} 2 & 7 & 5 \\ 6 & 20 & 10 \\ 4 & 3 & 0 \end{pmatrix}$$

```
{0, 4, 1}
```

```
{{{2, 7, 5}, {3, -1, -5}, {2, 11, 45}}, {1, 2, 3}, 0}
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 11 & 1 \end{pmatrix}$$

```
SparseArray[SummaryPanel[
  Specified elements: 6
  Dimensions: {3, 3}
  Default: 0
  Density: 0.667
  Elements:
  {1, 3} → 5
  {1, 2} → 7
  {1, 1} → 2
  {2, 2} → -1
  ⋮
]]
```

```
u
```

```
SparseArray[
  SummaryPanel[
    Specified elements: 6
    Dimensions: {3, 3}
    Default: 0
    Density: 0.667
    Elements:
    {1, 3} → 5
    {1, 2} → 7
    {1, 1} → 2
    {2, 2} → -1
    ⋮
  ]].u
```

```
LinearSolveFunction[
  SummaryPanel[ ⁻¹ Matrix dimensions: {3, 3}
                   Precision: ∞ ]]
```

$$\left\{-\frac{1}{3}, \frac{7}{9}, -\frac{43}{45}\right\}$$

```
{0, 4, 1}
```

Practical-8

Gauss Jacobi Method :

Question-1 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations x (0) = 0

$$4x + y + 2z = 4$$
$$-3x + 5y + z = 7$$
$$x + y + 3z = 3$$

```
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x = (4 - x2 - 2 x3) / 4, y = (7 + 3 x1 - x3) / 5, z = (3 - x1 - x2) / 3};
  i = i + 1; {x1 = x, x2 = y, x3 = z}; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	1.	1.4	1.
3	0.15	1.8	0.2
4	0.45	1.45	0.35
5	0.4625	1.6	0.36666667
6	0.41666667	1.6041667	0.3125
7	0.44270833	1.5875	0.32638889
8	0.43993056	1.6003472	0.32326389
9	0.43828125	1.5993056	0.31990741
10	0.44021991	1.5989873	0.3208044
11	0.43985098	1.5999711	0.32026427
12	0.4398751	1.5998577	0.32005932
13	0.44000591	1.5999132	0.32008906
14	0.43997717	1.5999857	0.32002697

Question-2 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations x (0) = 0

$$2x - y = 7$$

$$-x + 2y - z = 1$$
$$y + 2z = 1$$

```
ClearAll;
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x = (7 + x2) / 2, y = (1 + x1 + x3) / 2, z = (1 - x2) / 2};
  i = i + 1; {x1 = x, x2 = y, x3 = z}; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	3.5	0.5	0.5
3	3.75	2.5	0.25
4	4.75	2.5	-0.75
5	4.75	2.5	-0.75
6	4.75	2.5	-0.75
7	4.75	2.5	-0.75
8	4.75	2.5	-0.75
9	4.75	2.5	-0.75
10	4.75	2.5	-0.75
11	4.75	2.5	-0.75
12	4.75	2.5	-0.75
13	4.75	2.5	-0.75
14	4.75	2.5	-0.75

Question-3 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations $x(0) = 0$

$$10x + y + z = 12$$
$$2x + 10y + z = 13$$
$$2x + 2y + 10z = 14$$

```
ClearAll;
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x = (12 - x3 - x2) / 10, y = (13 - 2 x1 - x3) / 10, z = (14 - 2 x1 - 2 x2) / 10};
  i = i + 1; {x1 = x, x2 = y, x3 = z}; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	1.2	1.3	1.4
3	0.93	0.92	0.9
4	1.018	1.024	1.03
5	0.9946	0.9934	0.9916
6	1.0015	1.00192	1.0024
7	0.999568	0.99946	0.999316
8	1.0001224	1.0001548	1.0001944
9	0.99996508	0.99995608	0.99994456
10	1.0000099	1.0000125	1.0000158
11	0.99999717	0.99999644	0.99999551
12	1.0000008	1.000001	1.0000013
13	0.99999977	0.99999971	0.99999964
14	1.0000001	1.0000001	1.0000001

Question-4 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations x (0) = 0

$$x - 2y - 2z = 11$$
$$x + y - z = 0$$
$$2x + 2y + z = 3$$

```
ClearAll;
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x = (3 - x3 - 2 x2) / 2, y = (11 - x1 + 2 x3) / (-2), z = x1 + x2};
  i = i + 1; {x1 = x, x2 = y, x3 = z}; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	1.5	-5.5	0.
3	7.	-4.75	-4.
4	8.25	2.	2.25
5	-1.625	-3.625	10.25
6	0.	-16.5625	-5.25
7	20.6875	-0.25	-16.5625
8	10.03125	21.40625	20.4375
9	-30.125	-20.921875	31.4375
10	6.703125	-52.	-51.046875
11	79.023438	48.898438	-45.296875
12	-24.75	79.308594	127.92188
13	-141.76953	-145.79688	54.558594
14	120.01758	-130.94336	-287.56641

Practical-9

Gauss Seidel Iteration Method :

Question-1 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations x (0) = 0

$$4x + y + 2z = 4$$
$$-3x + 5y + z = 7$$
$$x + y + 2z = 3$$

```
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x1 = (4 - x2 - 2 x3) / 4, x2 = (7 + 3 x1 - x3) / 5, x3 = (3 - x1 - x2) / 2};
  i = i + 1; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	1.	2.	0.
3	0.5	1.7	0.4
4	0.375	1.545	0.54
5	0.34375	1.49825	0.579
6	0.3359375	1.4857625	0.58915
7	0.33398438	1.4825606	0.5917275
8	0.33349609	1.4817522	0.59237588
9	0.33337402	1.4815492	0.59253837
10	0.33334351	1.4814984	0.59257903
11	0.33333588	1.4814857	0.5925892
12	0.33333397	1.4814825	0.59259174
13	0.33333349	1.4814817	0.59259238
14	0.33333337	1.4814815	0.59259254

Question-2 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations x (0) = 0

$$2x - y = 7$$
$$-x + 2y - z = 1$$
$$y + 2z = 1$$

```
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x1 = (7 + x2) / 2, x2 = (1 + x1 + x3) / 2, x3 = (1 - x2) / 2};
  i = i + 1; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	3.5	2.25	-0.625
3	4.625	2.5	-0.75
4	4.75	2.5	-0.75
5	4.75	2.5	-0.75
6	4.75	2.5	-0.75
7	4.75	2.5	-0.75
8	4.75	2.5	-0.75
9	4.75	2.5	-0.75
10	4.75	2.5	-0.75
11	4.75	2.5	-0.75
12	4.75	2.5	-0.75
13	4.75	2.5	-0.75
14	4.75	2.5	-0.75

Question-3 : Perform 14 iterations upto 8 decimal places to solve the following system of linear equations x (0) = 0

$$x - 2y - 2z = 11$$

$$x + y - z = 0$$

$$2x + 2y + z = 3$$

Since the equations are not diagonally dominant, we interchange the equation to make them diagonally dominant.

```
x1 = 0.0; x2 = 0.0; x3 = 0.0;
i = 1;
output = {{x1, x2, x3}};
While[i < 14, {x1 = (3 - 2 x2 - x3) / 2, x2 = (11 - x1 + 2 x3) / (-2), x3 = x1 + x2};
  i = i + 1; output = Append[output, {x1, x2, x3}]];
Print[NumberForm[TableForm[output,
    TableHeadings → {Automatic, {"x1", "x2", "x3"}}], 8]];
```

	x1	x2	x3
1	0.	0.	0.
2	1.5	-4.75	-3.25
3	7.875	1.6875	9.5625
4	-4.96875	-17.546875	-22.515625
5	30.304688	32.167969	62.472656
6	-61.904297	-98.924805	-160.8291
7	180.83936	245.74878	426.58813
8	-457.54285	-660.85956	-1118.4024
9	1221.5608	1723.6828	2945.2435
10	-3194.8046	-4548.1458	-7742.9504
11	8421.121	11948.011	20369.132
12	-22131.077	-31440.17	-53571.247
13	58227.294	82679.394	140906.69
14	-153131.24	-217477.81	-370609.05

Practical – 10

Lagrange Interpotion :

Q1 : Given that f (1) = 2 ,

f (2) = 5 and f (3) = 10. Find the Lagrange Polynomial

of degree 2 using given data and also find f (2.5).

```
No = 3; sum = 0;
lagrange[No_, n_] :=
  Product[If[Equal[k, n], 1, (x - x[k]) / (x[n] - x[k])], {k, 1, No}];
For[i = 1, i ≤ No, i++, sum += (f[x[i]] * lagrange[No, i])];
Print[sum];
```

```
((120.018 - 120.018[2]) (120.018 - 120.018[3]) {2, 5, 10}[120.018[1]]) /
  ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) +
 ((120.018 - 120.018[1]) (120.018 - 120.018[3]) {2, 5, 10}[120.018[2]]) /
  ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) +
 ((120.018 - 120.018[1]) (120.018 - 120.018[2]) {2, 5, 10}[120.018[3]]) /
  ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3]))
```

```
sum = 0;
points = {{1, 2}, {2, 5}, {3, 10}};
No = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
lagrange[No_, n_] :=
  Product[If[Equal[k, n], 1, (x - x[k]) / (x[n] - x[k])], {k, 1, No}];
For[i = 1, i ≤ No, i++, sum += (f[x[i]] * lagrange[No, i])]
Expand[sum]
sum /. x → 2.5
```

```
3
```

```
{1, 2, 3}
```

```
{2, 5, 10}
```

```
(14404.2 {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) -
  (120.018 × 120.018[2] {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) -
  (120.018 × 120.018[3] {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) +
  (120.018[2] × 120.018[3] {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) +
  (14404.2 {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) -
  (120.018 × 120.018[1] {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) -
  (120.018 × 120.018[3] {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) +
  (120.018[1] × 120.018[3] {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) +
  (14404.2 {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3])) -
  (120.018 × 120.018[1] {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3])) -
  (120.018 × 120.018[2] {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3])) +
  (120.018[1] × 120.018[2] {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3]))
```

```
((2.5 - 2.5[2]) (2.5 - 2.5[3]) {2, 5, 10}[2.5[1]]) /
   ((2.5[1] - 2.5[2]) (2.5[1] - 2.5[3])) +
  ((2.5 - 2.5[1]) (2.5 - 2.5[3]) {2, 5, 10}[2.5[2]]) /
   ((-2.5[1] + 2.5[2]) (2.5[2] - 2.5[3])) +
  ((2.5 - 2.5[1]) (2.5 - 2.5[2]) {2, 5, 10}[2.5[3]]) /
   ((-2.5[1] + 2.5[3]) (-2.5[2] + 2.5[3]))
```

Q2 : Given that f (1) = 2 ,

f (3) = 5 and f (8) = 10. Find the Lagrange Polynomial

of degree 2 using given data and also find f (2.5).

```
sum = 0;
points = {{1, 2}, {3, 5}, {8, 10}};
No = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
lagrange[No_, n_] :=
  Product[If[Equal[k, n], 1, (x - x[k]) / (x[n] - x[k])], {k, 1, No}];
For[i = 1, i ≤ No, i++, sum += (f[x[i]] * lagrange[No, i])]
Expand[sum]
sum /. x → 3
```

```
3
```

```
{1, 3, 8}
```

```
{2, 5, 10}
```

```
(14 404.2 {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) -
 (120.018 × 120.018[2] {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) -
 (120.018 × 120.018[3] {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) +
 (120.018[2] × 120.018[3] {2, 5, 10}[120.018[1]]) /
   ((120.018[1] - 120.018[2]) (120.018[1] - 120.018[3])) +
 (14 404.2 {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) -
 (120.018 × 120.018[1] {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) -
 (120.018 × 120.018[3] {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) +
 (120.018[1] × 120.018[3] {2, 5, 10}[120.018[2]]) /
   ((-120.018[1] + 120.018[2]) (120.018[2] - 120.018[3])) +
 (14 404.2 {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3])) -
 (120.018 × 120.018[1] {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3])) -
 (120.018 × 120.018[2] {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3])) +
 (120.018[1] × 120.018[2] {2, 5, 10}[120.018[3]]) /
   ((-120.018[1] + 120.018[3]) (-120.018[2] + 120.018[3]))
```

$$\frac{(3-3[2])\,(3-3[3])\,\{2,5,10\}[3[1]]}{(3[1]-3[2])\,(3[1]-3[3])}+\frac{(3-3[1])\,(3-3[3])\,\{2,5,10\}[3[2]]}{(-3[1]+3[2])\,(3[2]-3[3])}+\frac{(3-3[1])\,(3-3[2])\,\{2,5,10\}[3[3]]}{(-3[1]+3[3])\,(-3[2]+3[3])}$$

Practical-11

Newton Interpolation :

Q1: Given that f(3) = 293, f(5) = 508, f(6) = 585, f(9) = 764. Find the unique Polynomial of degree 2 or less using given data and also find f(2.5).

```
sum = 0;
points = {{3, 293}, {5, 508}, {6, 585}, {9, 764}};
n = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
dd[k_] :=
 Sum[(f[[i]] / Product[If[Equal[j, i], 1, (y[[i]] - y[[j]])], {j, 1, k}]), {i, 1, k}]
p[x_] = Sum[(dd[i] * Product[If[i ≤ j, 1, x - y[[j]]], {j, 1, i - 1}]), {i, 1, n}]
Simplify[p[x]]
Evaluate[p[2.5]]
```

4

{3, 5, 6, 9}

{293, 508, 585, 764}

$$293 + \frac{215}{2}(-3 + x) - \frac{61}{6}(-5 + x)(-3 + x) + \frac{35}{36}(-6 + x)(-5 + x)(-3 + x)$$

$$\frac{1}{36}\left(-9702 + 9003\,x - 856\,x^2 + 35\,x^3\right)$$

222.288

Q2 : Given that f (5) = 295 f (7) = 510 f (8) = 587 f (11) = 7646. Find the unique Polynomial of degree 2 or less using given data and also find f(2.5).

```
sum = 0;
points = {{5, 295}, {7, 510}, {8, 587}, {11, 766}};
n = Length[points]
y = points[[All, 1]]
f = points[[All, 2]]
dd[k_] :=
 Sum[(f[[i]] / Product[If[Equal[j, i], 1, (y[[i]] - y[[j]])], {j, 1, k}]), {i, 1, k}]
p[x_] = Sum[(dd[i] * Product[If[i ≤ j, 1, x - y[[j]]], {j, 1, i - 1}]), {i, 1, n}]
Simplify[p[x]]
Evaluate[p[2.5]]
```

4

{5, 7, 8, 11}

{295, 510, 587, 766}

$$295 + \frac{215}{2}(-5 + x) - \frac{61}{6}(-7 + x)(-5 + x) + \frac{35}{36}(-8 + x)(-7 + x)(-5 + x)$$

$$\frac{1}{36}\left(-31340 + 12847 x - 1066 x^2 + 35 x^3\right)$$

-148.281

Practical – 12

Simpson Rule :

Ques tion1 : – Evalute the integral $I = \int \frac{1}{x} \, dx$ using " Simpson Method "

```
a = Input["Enter the left end point:"];
b = Input["Enter the right end point:"];
n = Input["Enter the number of sub intervals to be formed:"]
h = (b - a) / n;
y = Table[a + i h, {i, 1, n}];
f[x] := 1 / x;
sumodd = 0;
sumeven = 0;
For[i = 1, i < n, i += 2, sumodd += 4 * f[x] /. x → y[[i]]];
For[i = 2, i < n, i += 2, sumodd += 2 * f[x] /. x → y[[i]]];
Sn = (h / 3) * ((f[x] /. x → a) + N[sumodd] + N[sumeven] + (f[x] /. x → b));
Print["For n=", n, ",Simpson estimate is:", Sn]
in = Integrate[1 / x, {x, 1, 2}]
Print["True value is", in]
Print["absolute error is ", Abs[Sn - in]]
```

6

For n=6,Simpson estimate is: $\frac{1}{18}$
(0. + {2, 5, 10}[1] + {2, 5, 10}[2] + 4. {2., 5., 10.}[1.16667] + 2. {2., 5., 10.}[1.33333] +
4. {2., 5., 10.}[1.5] + 2. {2., 5., 10.}[1.66667] + 4. {2., 5., 10.}[1.83333])

$$\int_1^2 0.00833211\, d120.018$$

True value is $\int_1^2 0.00833211\, d120.018$

absolute error is Abs$\Big[-\int_1^2 0.00833211\, d120.018 +$
$\frac{1}{18}$ (0. + {2, 5, 10}[1] + {2, 5, 10}[2] + 4. {2., 5., 10.}[1.16667] + 2. {2., 5., 10.}[1.33333] +
4. {2., 5., 10.}[1.5] + 2. {2., 5., 10.}[1.66667] + 4. {2., 5., 10.}[1.83333])$\Big]$

Ques tion2 : – Evalute the integral I =

$\int \text{Cos}(x)\, dx$ using " Simpson Method "

```
a = Input["Enter the left end point:"];
b = Input["Enter the right end point:"];
n = Input["Enter the number of sub intervals to be formed:"]
h = (b - a) / n;
y = Table[a + i h, {i, 1, n}];
f[x] := Cos[x];
sumodd = 0;
sumeven = 0;
For[i = 1, i < n, i += 2, sumodd += 4 * f[x] /. x → y[[i]]];
For[i = 2, i < n, i += 2, sumodd += 2 * f[x] /. x → y[[i]]];
Sn = (h / 3) * ((f[x] /. x → a) + N[sumodd] + N[sumeven] + (f[x] /. x → b));
Print["For n=", n, ",Simpson estimate is:", Sn]
in = Integrate[Cos[x], {x, 1, 2}]
Print["True value is", in]
Print["absolute error is ", Abs[Sn - in]]
```

```
6
```

For n=6,Simpson estimate is:0.841475

```
-Sin[1] + Sin[2]
```

True value is-Sin[1] + Sin[2]

absolute error is 0.773648