EXPERIMENT-7

AIM-To find the solution of equation using simpson's 3/8th rule

THEORY-

Simpson's 3/8th Rule is a Numerical technique to find the definite integral of a function within a given interval.
It's so called because the value 3/8 appears in the formula.
The function is divided into many sub-intervals and each interval is approximated by a cubic curve. And the area is then calculated to find the integral. The more is the number of sub-intervals used, the better is the approximation.

FORMULA USED-

$$\int_a^b f(x)dx = \frac{3h}{8}\left(\mathrm{F(a)} + 3\mathrm{F}\left(\frac{2a+b}{3}\right) + 3\mathrm{F}\left(\frac{a+2b}{3}\right) + \mathrm{F(b)}\right)$$

Where h = (b-a)/2
f(x) is called the integrand
a= lower limit of integration
b= upper limit of integration

ALGORITHM-

1. Given a function f(x):
2. (Get user inputs)
   Input
   a,b=endpoints of interval
   n=number of intervals(Even)
   (Do the integration)
3. Set h= (b-a)/n.
4. Set sum=0.
5. Begin For i= 1 to n -1
Set x =a + h*i.
If i%3=0
Then Set sum=sum+2*f(x)
Else
Set sum=sum+3*f(x)
End For
6. Set sum = sum + f(a)+f(b)

EXPERIMENT-7

7. Set ans = sum*(3h/8).
8. End

CODE-

```
#include<iostream>
using namespace std;
// Given function to be integrated
float func( float x)
{
return (1 / (1 + x));
}
// Function to perform calculations
float calculate(float lower_limit, float upper_limit,
int interval_limit )
{
float value;
float interval_size = (upper_limit - lower_limit)
/ interval_limit;
float sum = func(lower_limit) + func(upper_limit);
// Calculates value till integral limit
for (int i = 1 ; i < interval_limit ; i++)
{
if (i % 3 == 0)
sum = sum + 2 * func(lower_limit + i * interval_size);
else
sum = sum + 3 * func(lower_limit + i * interval_size);
}
return ( 3 * interval_size / 8 ) * sum ;
}
// Driver Code
int main()
{
int interval_limit = 6;
float lower_limit = 0;
float upper_limit = 3;
cout<<"\nupper limit = "<<upper_limit;
cout<<"\nlower limit = "<<lower_limit;
cout<<"\nintervals = "<<interval_limit;
float integral_res = calculate(lower_limit, upper_limit,interval_limit);
cout <<"\n\nthe answer = "<<integral_res<<"\n";
return 0;
```

```
}
```

OUTPUT-