STUDY OF THE NUMERICAL INTEGRATION RULES USING C-LANGUAGE

A Project Report

Submitted by

GADDE ANUSRI (Y193099019)

J SIREESHA (Y193099020)

SHAIK HAFIJA (Y193099024)

Under the guidanceof

G. Sowjanya M.Sc.,

Lecturer

Department of Mathematics

SGK GOVERNMENT DEGREE COLLEGE

VINUKONDA

SGK GOVERNMENT DEGREE COLLEGE

VINUKONDA

BONAFIDE CERTIFICATE

Certified that this project STUDY OF THE NUMERICAL INTEGRATION RULES USING C-PROGRAM is the bonafide work of GADDE ANUSRI (Y193099019), JEEDIMALLA SIREESHA (Y193099020) and SHAIK HAFIJA (Y193099024) who carried out the project work under my supervision.

Supervisor Principal

PROJECT APPROVAL SHEET

Following team has done the appropriate work related to the THE STUDY OF NUMERICAL INTEGRATION RULES USING C-PROGRAM and is being submitted to SGK GOVERNMENT DEGREE COLLEGE, VINUKONDA-522647.

Team:

GADDE ANUSRI	(Y193099019)
J SIREESHA	(Y193099020)
SHAIK HAFIJA	(Y193099024)

Supervisor:

External Examiner:

Date:

Place: SGK GOVERNMENT DEGREE COLLEGE, VINUKONDA-522647.

TABLE OF CONTENTS

CHAPTER 1

NEWTON-COTE'S QUADRATURE FORMULA

INTRODUCTION

To find the definite integral, usually we use the fundamental theorem of calculus, where we have to apply the antiderivative techniques of integration. However, sometimes, it isn't easy to find the antiderivative of an integral, like in Scientific experiments, where the function has to be determined from the observed readings. Therefore, numerical methods are used to approximate the integral in such conditions.

We know that a definite integral of the form $\int_a^b f(x)dx$ represents the area under the curve $y = f(x)$, enclosed between the limits $x = a$ and $x = b$. This integration is possible only if $f(x)$ is explicitly given and if it is integrable. The problem of numerical integration can be stated as follows:

Given a set of $(n + 1)$ data points $(x_i, y_i), i = 0,1,2, \dots, n$ of the function $y = f(x)$, where $f(x)$ is not known explicitly, it is required to evaluate $\int_{x_0}^{x_n} f(x)dx$.

The problem of numerical integration, like that of numerical differentiation is solved by replacing $f(x)$ with an interpolating polynomial $P_n(x)$ and obtaining $\int_{x_0}^{x_n} P_n(x)dx$ which is approximately taken as the value of $\int_{x_0}^{x_n} f(x)dx$. Numerical Integration is also known as Numerical Quadrature.

NEWTON-COTE'S QUADRATURE FORMULA

This is also known as General Quadrature formula and is the most popular and widely used numerical integration formula. It forms the basis for a number of numerical integration methods known as Newton-Cote's methods.

Derivation of Newton-Cotes formula

Let the interval $[a, b]$ be divided into n equal subintervals with interval of differencing h such that $a = x_0 < x_1 < x_2 < \cdots\cdots < x_n = b$. Then $x_n = x_0 + nh$.

Newton's forward interpolation formula is

$$y(x) = y(x_0 + uh) = P_n(x)$$

$$= y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \cdots\cdots\cdots \qquad (1)$$

where $u = \frac{x-x_0}{h}$.

Now, instead of $f(x)$, replace it by this interpolating polynomial.

$\therefore \int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_n} P_n(x)dx$ where $P_n(x)$ is an interpolating polynomial of degree n.

$$= \int_{x_0}^{x_0+nh} P_n(x)dx$$

$$= \int_{x_0}^{x_0+nh} [y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \cdots\cdots\cdots] dx$$

Since $x_n = x_0 + uh$, we have $dx = h.dp$ and the integration limits changes from 0 to n. Therefore, the above integral becomes

$$\int_{x_0}^{x_n} f(x)dx = h\int_0^n [y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \cdots]\,du$$

$$= h\int_0^n [y_0 + u\Delta y_0 + \frac{u^2-1}{2!}\Delta^2 y_0 + \frac{u^3-3u^2+2u}{3!}\Delta^3 y_0 + \cdots]\,du$$

$$= h\,[uy_0 + \frac{u^2}{2}\Delta y_0 + (\frac{u^3}{6} - \frac{u^2}{4})\Delta^2 y_0 + (\frac{u^4}{24} - \frac{u^3}{6} + \frac{u^2}{6})\Delta^3 y_0 + \cdots]_0^n$$

$$= h\,[ny_0 + \frac{n^2}{2}\Delta y_0 + (\frac{n^3}{6} - \frac{n^2}{4})\Delta^2 y_0 + (\frac{n^4}{24} - \frac{n^3}{6} + \frac{n^2}{6})\Delta^3 y_0 + (\frac{n^5}{120} - \frac{3n^4}{48} + \frac{11n^3}{72} - \frac{n^2}{8})\Delta^4 y_0 \cdots] \qquad (2)$$

This is called Newton-Cote's Quadrature formula. From this general formula, we get different integration formulae by putting $n = 1,2,3,\cdots$

With $n = 1$, we get $Trapezoidal\ rule$

With $n = 2$, we get $Simpson's\ \frac{1}{3}\ rule$

With $n = 3$, we get $Simpson's\ \frac{3}{8}\ rule$

With $n = 4$, we get $Boole's\ rule$

With $n = 6$, we get $Weddle's\ rule$

Note: The software used for c-program is Code Blocks.

CHAPTER 2

TRAPEZOIDAL RULE WITH C - PROGRAM

Trapezoidal Rule is one of the important integration rules. The name trapezoidal is because when the area under the curve is evaluated, then the total area is divided into small trapezoids instead of rectangles. This rule is used for approximating the definite integrals where it uses the linear approximations of the functions.

Here the function $f(x)$ is approximated by a first-order polynomial $P_1(x)$ which passes through two points.

Taking $n = 1$, in the General Quadrature formula, all differences higher than the first order will become zero and thus we get

$$\int_{x_0}^{x_1} f(x)dx = \int_{x_0}^{x_0+h} f(x)dx = h\left[y_0 + \frac{1}{2}\Delta y_0\right] = h\left[y_0 + \frac{1}{2}(y_1 - y_0)\right] = \frac{h}{2}(y_0 + y_1)$$

Similarly,

$$\int_{x_1}^{x_2} f(x)dx = \int_{x_0+h}^{x_0+2h} f(x)dx = h\left[y_1 + \frac{1}{2}\Delta y_1\right] = h\left[y_1 + \frac{1}{2}(y_2 - y_1)\right] = \frac{h}{2}(y_1 + y_2)$$

$$\int_{x_2}^{x_3} f(x)dx = \int_{x_0+h}^{x_0+2h} f(x)dx = \frac{h}{2}(y_{n-1} + y_n)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

Finally, $\int_{x_0+(n-1)h}^{x_0+nh} f(x)dx = \frac{h}{2}(y_{n-1} + y_n)$

$$\therefore \int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \cdots + \int_{x_0+(n-1)h}^{x_0+nh} f(x)dx$$

$$= \frac{h}{2}(y_0 + y_1) + \frac{h}{2}(y_1 + y_2) + \cdots + \frac{h}{2}(y_{n-1} + y_n)$$

$$= \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + y_4 + \cdots + y_{n-1})]$$

$$\boldsymbol{\int_{x_0}^{x_n} f(x)dx = \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + y_4 + \cdots + y_{n-1})]}$$

$$OR$$

$$\int_{x_0}^{x_n} f(x)dx = \frac{h}{2}[(\text{Sum of the First and Last Ordinates})$$

$$+2(\text{Sum of the Remaining Ordinates})]$$

This is known as Trapezoidal Rule.

Geometrical Interpretation

Consider the points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $\cdots \cdots$, $P_n(x_n, y_n)$. Suppose the curve $y = f(x)$ passing through the above points be approximated by the union of the line segments joining $(P_0, P_1), (P_1, P_2), (P_2, P_3), \cdots \cdots, (P_{n-1}, P_n)$.

Geometrically, the curve $y = f(x)$ is replaced by n straight line segments joining the points $P_0(x_0, y_0)$ and $P_1(x_1, y_1)$; $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$; $\cdots\cdots$ $P_{n-1}(x_{n-1}, y_{n-1})$ and $P_n(x_n, y_n)$. The area bounded by the curve $y = f(x)$, $x - axis$ and the ordinates $x = x_0$ and $x = x_n$ is then approximately equal to the sum of the areas of the n trapeziums.

Though this method is very simple for calculation purposes of numerical integration, the error in this case is significant. The accuracy of the result can be improved by increasing the number of intervals or by decreasing the value of h.

Example Evaluate $\int_0^1 \frac{1}{1+x} dx$ by using Trapezoidal rule with six sub-intervals.

Solution Divide the interval $[0,1]$ into six subintervals.

Here $h = \frac{b-a}{n} = \frac{1-0}{6} = \frac{1}{6}$

The values of x and y are tabulated as below:

x	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1
$y = \frac{1}{1+x}$	1	0.857142	0.75	0.66667	0.6	0.545454	0.5

$\therefore y_0 = 1, y_1 = 0.8571, y_2 = 0.75, y_3 = 0.6667, y_4 = 0.6, y_5 = 0.5454, y_6 = 0.5$

By Trapezoidal Rule,

$$\int_0^1 \frac{1}{1+x} dx = \frac{h}{2}[(y_0 + y_6) + 2(y_1 + y_2 + y_3 + y_4 + y_5)]$$

$$= \frac{1}{12}[(1 + 0.5) + 2(0.857142 + 0.75 + 0.666667 + 0.6 + 0.545454)]$$

$$= 0.694877$$

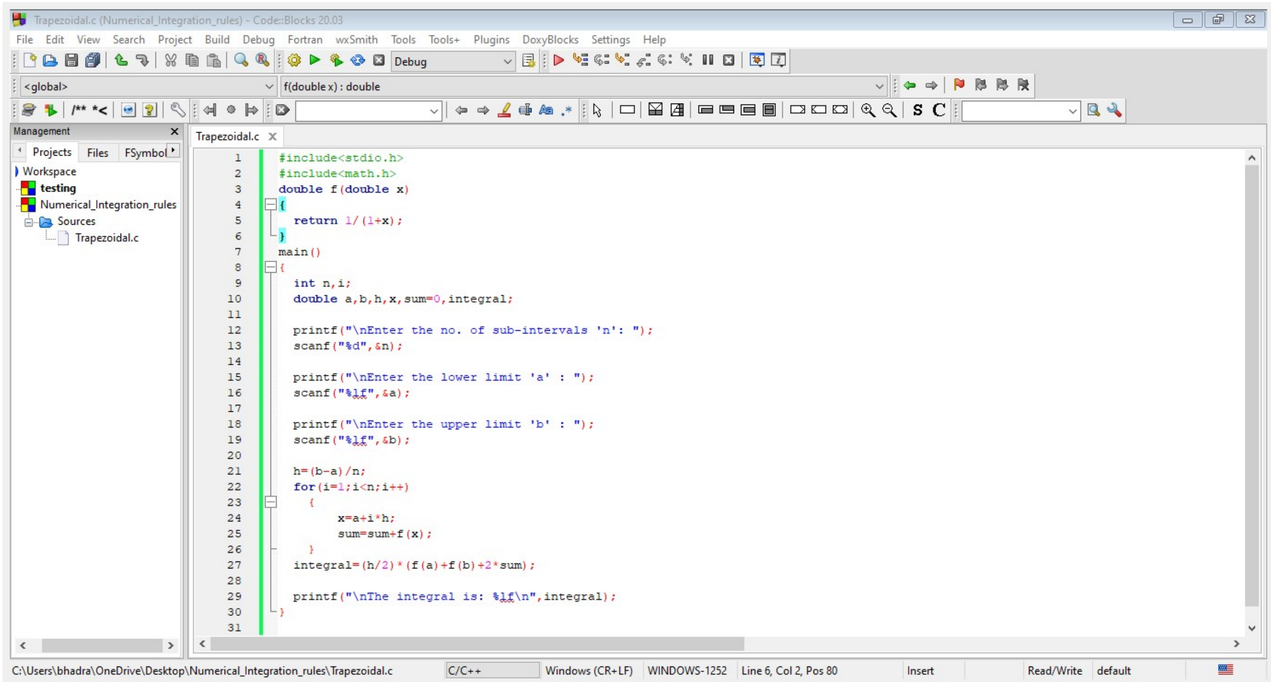
Analytical Solution: $\int_0^1 \frac{1}{1+x} dx = [\log(1+x)]_{\mathbf{0}}^{\mathbf{1}} = ln2 = 0.693147$

TRAPEZOIDAL RULE C-PROGRAM

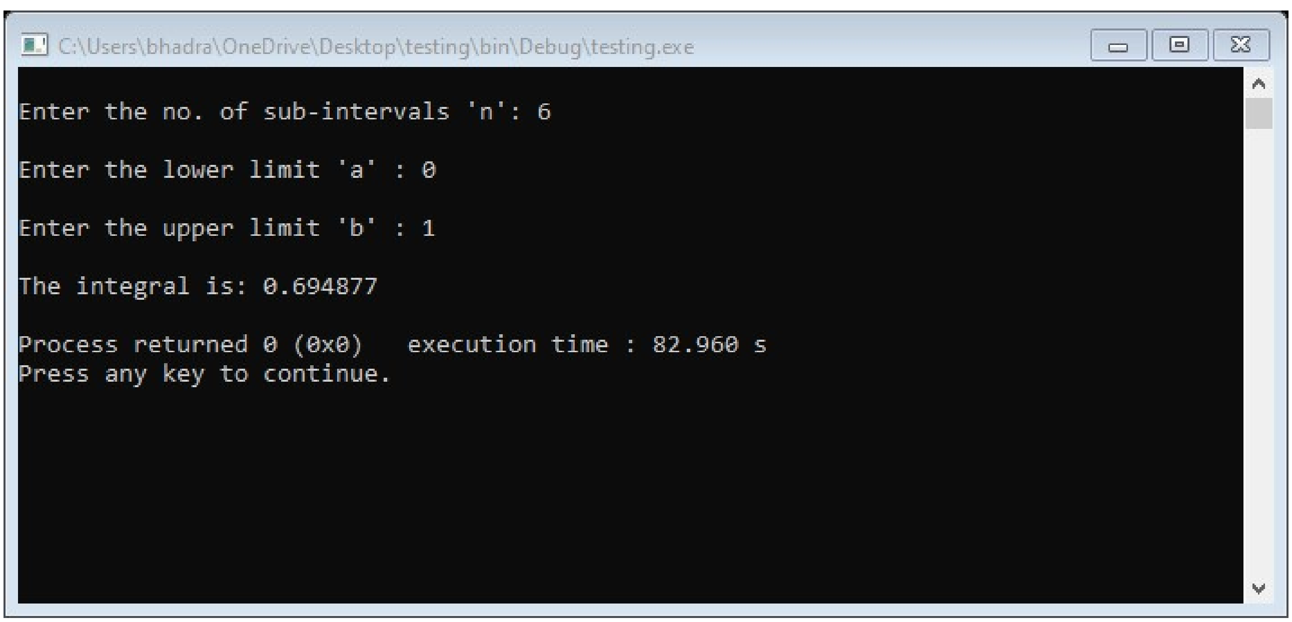
```
#include<stdio.h>
#include<math.h>

double f(double x)
{
  return 1/(1+x);
}

main()
{
  int n,i;
  double a,b,h,x,sum=0,integral;

  printf("\nEnter the no. of sub-intervals 'n': ");
  scanf("%d",&n);

  printf("\nEnter the lower limit 'a': ");
  scanf("%lf",&a);

  printf("\nEnter the upper limit 'b': ");
  scanf("%lf",&b);

  h=(b-a)/n;
  for(i=1;i<n;i++)
     {
          x=a+i*h;
          sum=sum+f(x);
     }
  integral=(h/2)*(f(a)+f(b)+2*sum);

  printf("\nThe integral is: %lf\n",integral);
}
```

===================== END OF THE PROGRAM ========================

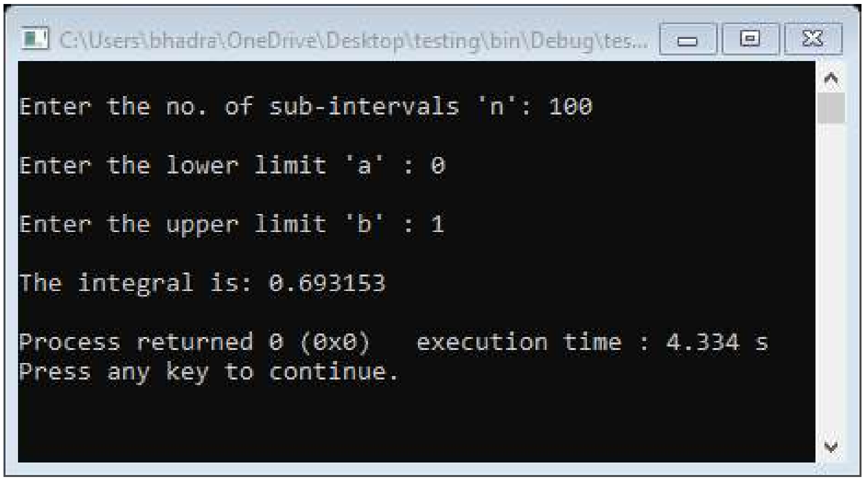
The screeshot of the c program for trapezoidal rule:

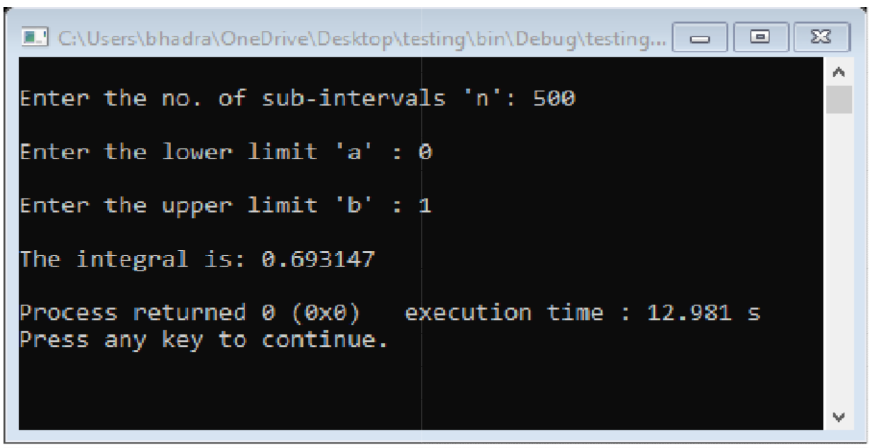


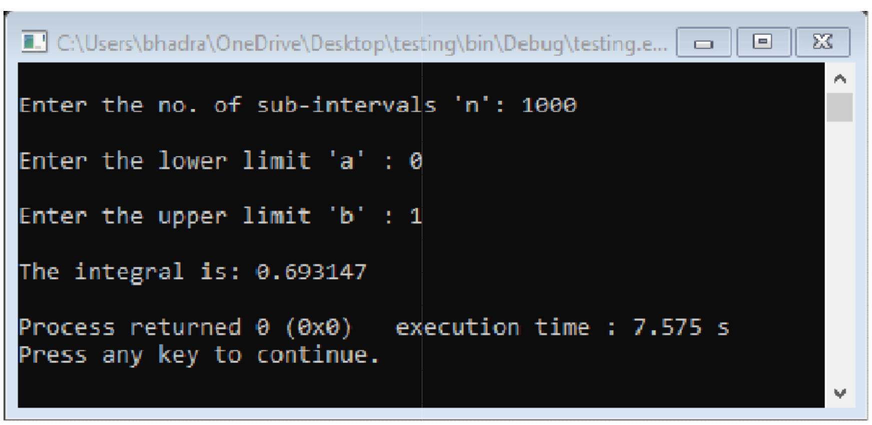
The screeshot of the output with $n = 6, a = 0, b = 1$ which gives the integral 0.694877, which matches exactly with the answer in the above example:



From the below output screenshots, we can conclude that the exact value of the integral is 0.693147 which is obtained by increasing the number of sub-intervals.







CHAPTER 3

SIMPSON'S RULES WITH C - PROGRAMS

We have Simpson's $\frac{1}{3}-$rule and Simpson's $\frac{3}{8}-$rule.

Simpson's $\frac{1}{3}-$rule

In Simpson's $\frac{1}{3}-$rule, the function $f(x)$ is approximated by a second order polynomial $P_2(x)$ which passes through three successive points.

Taking $n = 2$ in the the General Quadrature formula, by replacing the curve $y = f(x)$ by $\frac{n}{2}$ parabolas, all differences higher than the second order will become zero and thus we get

$$\int_{x_0}^{x_2} f(x)dx = \int_{x_0}^{x_0+2h} f(x)dx = 2h\,[y_0 + \Delta y_0 + \frac{1}{6}\Delta^2 y_0]$$

$$= 2h\,[y_0 + (y_1 - y_0) + \frac{1}{6}(y_2 - 2y_1 + y_0)]$$

$$= \frac{h}{3}(y_0 + 4y_1 + y_2)$$

Similarly,

$$\int_{x_2}^{x_4} f(x)dx = \frac{h}{3}(y_2 + 4y_3 + y_4)$$

$$\int_{x_4}^{x_6} f(x)dx = \frac{h}{3}(y_4 + 4y_5 + y_6)$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$\int_{x_{n-2}}^{x_n} f(x)dx = \frac{h}{3}(y_{n-2} + 4y_{n-1} + y_n)$$

Adding all these integrals, we get

$$\int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \int_{x_4}^{x_6} f(x)dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)dx$$

$$= \frac{h}{3}(y_0 + 4y_1 + y_2) + \frac{h}{3}(y_2 + 4y_3 + y_4) + \cdots + \frac{h}{3}(y_{n-2} + 4y_{n-1} + y_n)$$

$$= \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2})]$$

$$\boldsymbol{\int_{x_0}^{x_n} f(x)dx = \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2})]}$$

$$OR$$

$$\int_{x_0}^{x_n} f(x)dx = \frac{h}{3}[(\text{Sum of the First and Last Ordinates})$$

$$+4(\text{Sum of the Odd Ordinates}) + 2(\text{Sum of the remaining Even Ordinates})]$$

with the convention that $y_1, y_3, \dots, y_{n-1}$ are odd ordinates and $y_0, y_2, y_4, \dots, y_{n-2}, y_n$ are even ordinates.

This is known as $Simpsons\ \frac{1}{3} - rule$. The number of intervals in this rule must be even.

Example Evaluate $\int_0^1 \frac{1}{1+x} dx$ by using Simpson's $\frac{1}{3}$—rule with six sub-intervals.

Solution Divide the interval $[0,1]$ into six subintervals.

Here $h = \frac{b-a}{n} = \frac{1-0}{6} = \frac{1}{6}$

The values of x and y are tabulated as below:

x	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1
$y = \frac{1}{1+x}$	1	0.857142	0.75	0.66667	0.6	0.545454	0.5

$\therefore y_0 = 1, y_1 = 0.857142, y_2 = 0.75, y_3 = 0.6667, y_4 = 0.6, y_5 = 0.545454, y_6 = 0.5$

By Simpson's $\frac{1}{3}$ —rule,

$$\int_0^1 \frac{1}{1+x} dx = \frac{h}{3}[(y_0 + y_6) + 4(y_1 + y_3 + y_5) + 2(y_2 + y_4)]$$

$$= \frac{1}{18}[(1 + 0.5) + 4(0.857142 + 0.666667 + 0.545454) + 2(0.75 + 0.6)]$$

$$= 0.69316956$$

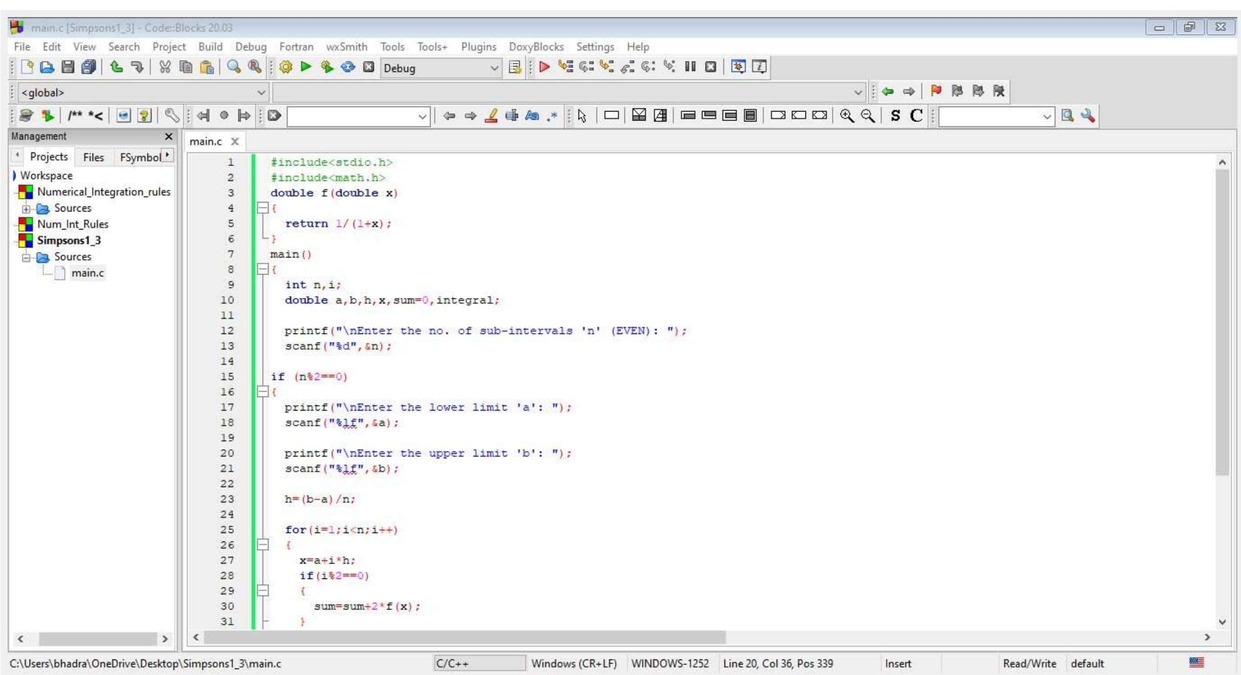
SIMPSON’S ONE-THIRD RULE C-PROGRAM

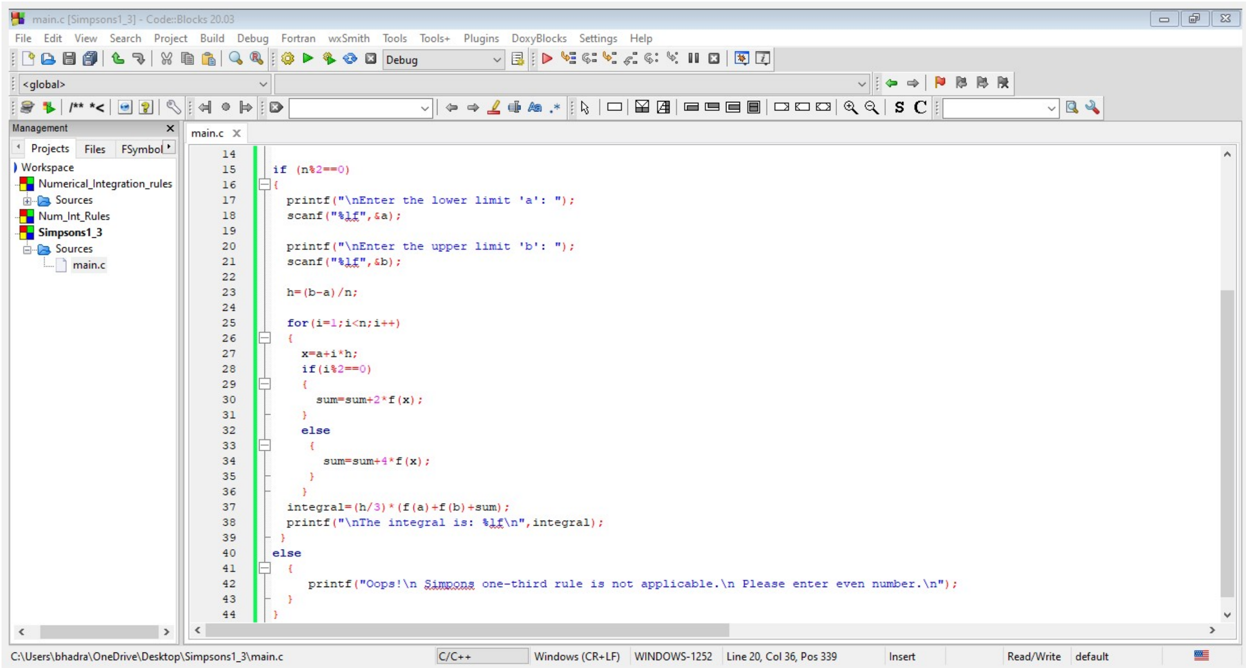
```
#include<stdio.h>
#include<math.h>

double f(double x)
{
  return 1/(1+x);
}

main()
{
  int n,i;
  double a,b,h,x,sum=0,integral;

  printf("\nEnter the no. of sub-intervals ‘n’ (EVEN): ");
  scanf("%d",&n);

if (n%2==0)
{
  printf("\nEnter the lower limit ‘a’: ");
  scanf("%lf",&a);

  printf("\nEnter the upper limit ‘b’: ");
  scanf("%lf",&b);

  h=(b-a)/n;

  for(i=1;i<n;i++)
  {
    x=a+i*h;
    if(i%2==0)
    {
      sum=sum+2*f(x);
    }
```

```
    else
     {
       sum=sum+4*f(x);
     }
    }
  integral=(h/3)*(f(a)+f(b)+sum);

  printf("\nThe integral is: %lf\n",integral);

 }

else
  {
     printf("Oops!\n Simpons one-third rule is not applicable.\n
Please enter even number.\n");
   }
}
```

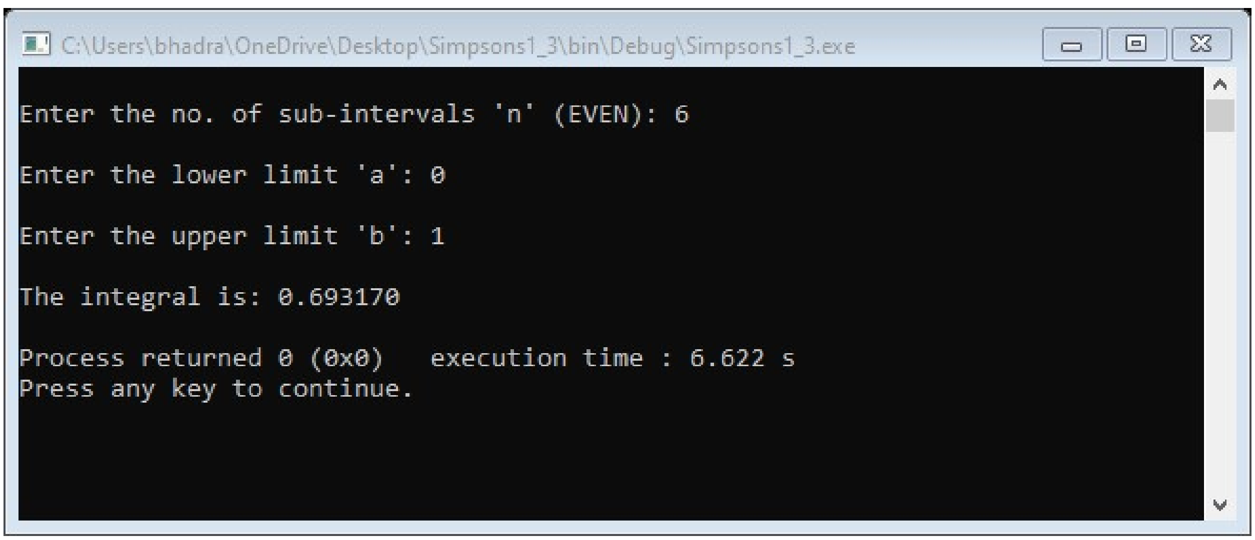
================================ END OF THE PROGRAM ================================

The screeshots of the c program for Simpson's $\frac{1}{3}-$ rule in the CodeBlocks software:

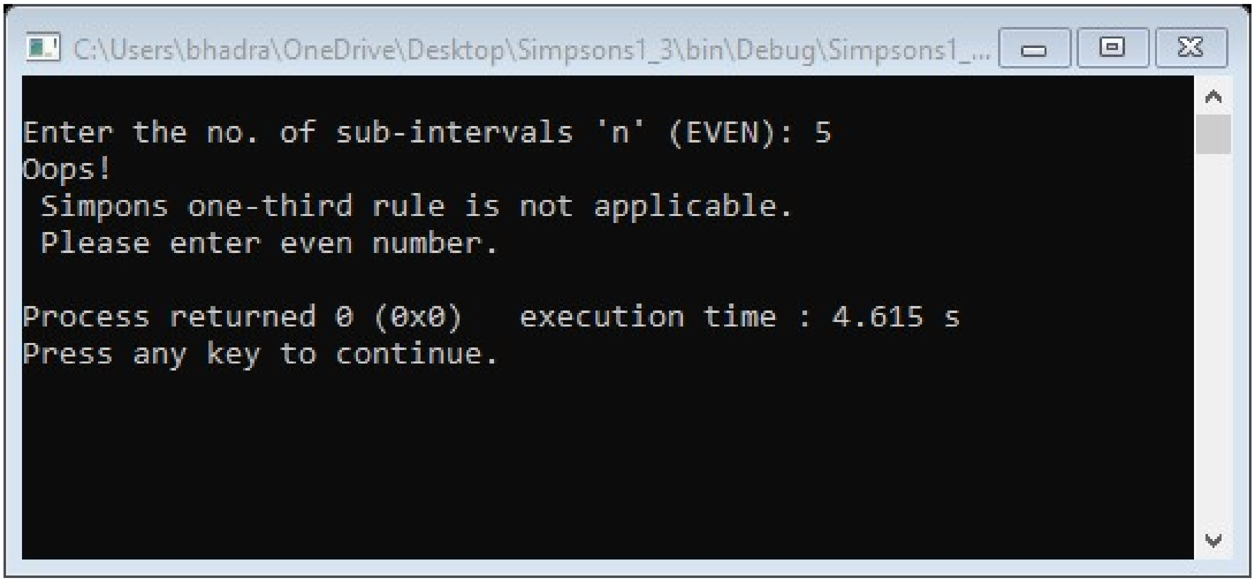




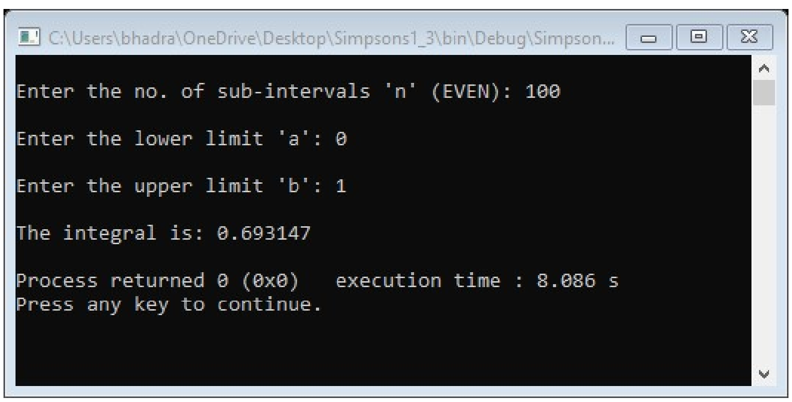
The integral value with $n = 6$, showing the output:

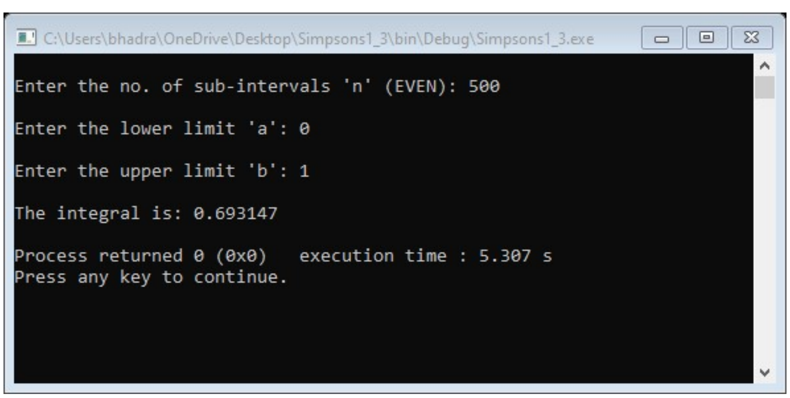


Simpson's $\frac{1}{3}$−rule is not applicable if n is not an even number:



From the below output screenshots, we can conclude that the exact value of the integral is 0.693147 which is obtained by increasing the number of sub-intervals.





Simpson's $\frac{3}{8}$ −rule

In Simpson's $\frac{3}{8}$−rule, the function $f(x)$ is approximated by a third order polynomial $P_3(x)$ which passes through four successive points.

Taking $n = 3$ in the the General Quadrature formula, all differences higher than the third order will become zero and thus we get

$$\int_{x_0}^{x_3} f(x)dx = 3h\left[y_0 + \frac{3}{2}\Delta y_0 + \frac{3}{4}\Delta^2 y_0 + \frac{1}{8}\Delta^3 y_0\right]$$

$$= 3h\left[y_0 + \frac{3}{2}(y_1 - y_0) + \frac{3}{4}(y_2 - 2y_1 + y_0) + \frac{1}{8}(y_3 - 3y_2 + 3y_1 - y_0)\right]$$

$$= \frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3)$$

Similarly, $\int_{x_3}^{x_6} f(x)dx = \frac{3h}{8}(y_3 + 3y_4 + 3y_5 + y_6)$

$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$

$$\int_{x_{n-3}}^{x_n} f(x)dx = \frac{3h}{8}(y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n)$$

Adding all these integrals, we get

$$\int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_3} f(x)dx + \int_{x_3}^{x_6} f(x)dx + \cdots + \int_{x_{n-3}}^{x_n} f(x)dx$$

$$= \frac{3h}{8}[(y_0 + y_n) + 3(y_1 + y_2 + y_4 + y_5 + \cdots + y_{n-1})$$

$$+2(y_3 + y_6 + \cdots + y_{n-3})]$$

$$\int_{x_0}^{x_n} f(x)dx = \frac{3h}{8}[(y_0 + y_n) + 3(y_1 + y_2 + y_4 + y_5 + \cdots + y_{n-1})$$

$$+ 2(y_3 + y_6 + \cdots + y_{n-3})]$$

This is known as $Simpsons\ \frac{3}{8} - rule$. The number of intervals in this rule must a multiple of 3.

Example Evaluate $\int_0^1 \frac{1}{1+x} dx$ by using Simpson's $\frac{3}{8}$—rule with six sub-intervals.

Solution Divide the interval $[0,1]$ into six subintervals.

Here $h = \frac{b-a}{n} = \frac{1-0}{6} = \frac{1}{6}$

The values of x and y are tabulated as below:

x	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1
$y = \frac{1}{1+x}$	1	0.857142	0.75	0.666667	0.6	0.545454	0.5

$\therefore y_0 = 1, y_1 = 0.857142, y_2 = 0.75, y_3 = 0.666667, y_4 = 0.6, y_5 = 0.545454, y_6 = 0.5$

By Simpson's $\frac{3}{8}$ —rule, $\int_0^1 \frac{1}{1+x} dx = \frac{3h}{8}[(y_0 + y_6) + 3(y_1 + y_2 + y_4 + y_5) + 2(y_3)]$

$$= \frac{3}{48}[(1 + 0.5) + 3(0.857142 + 0.75 + 0.6 + 0.545454) + 2(0.666667)]$$

$$= 0.69319513$$

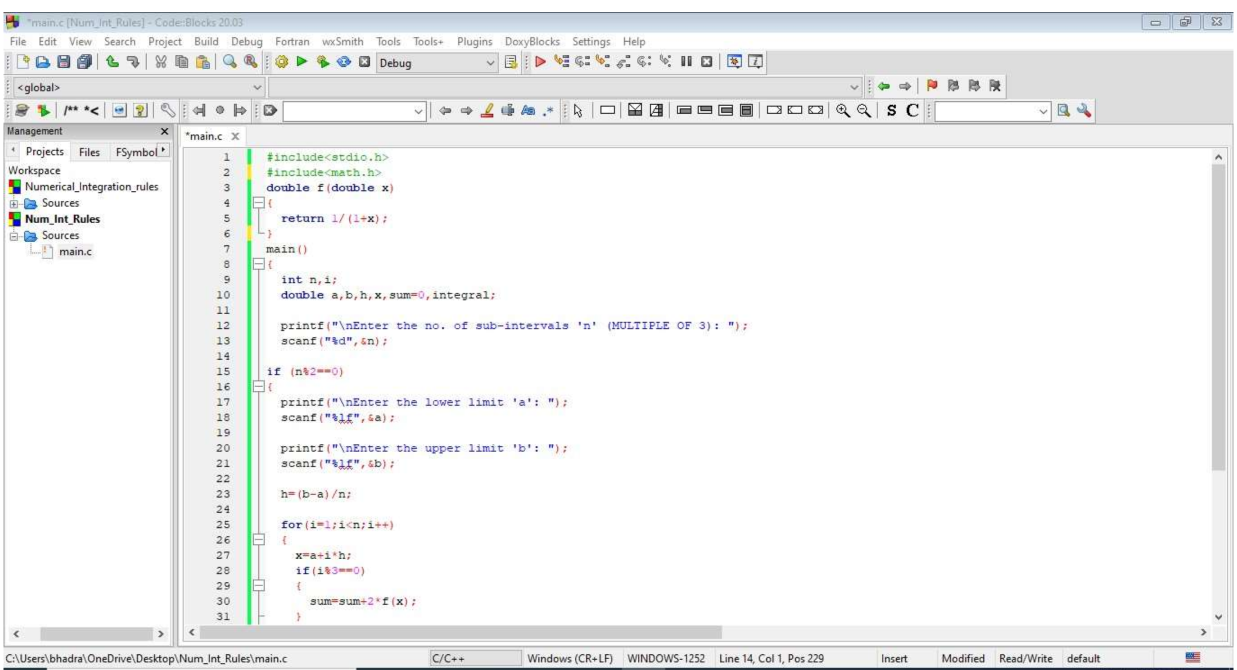
SIMPSON’S THREE-EIGHT RULE C-PROGRAM

```
#include<stdio.h>
#include<math.h>

double f(double x)
{
  return 1/(1+x);
}

main()
{
  int n,i;
  double a,b,h,x,sum=0,integral;

  printf("\nEnter the no. of sub-intervals ‘n’ (EVEN): ");
  scanf("%d",&n);

if (n%3==0)
{
  printf("\nEnter the lower limit ‘a’: ");
  scanf("%lf",&a);

  printf("\nEnter the upper limit ‘b’: ");
  scanf("%lf",&b);

  h=(b-a)/n;

  for(i=1;i<n;i++)
  {
    x=a+i*h;
    if(i%3==0)
    {
      sum=sum+2*f(x);
    }
```

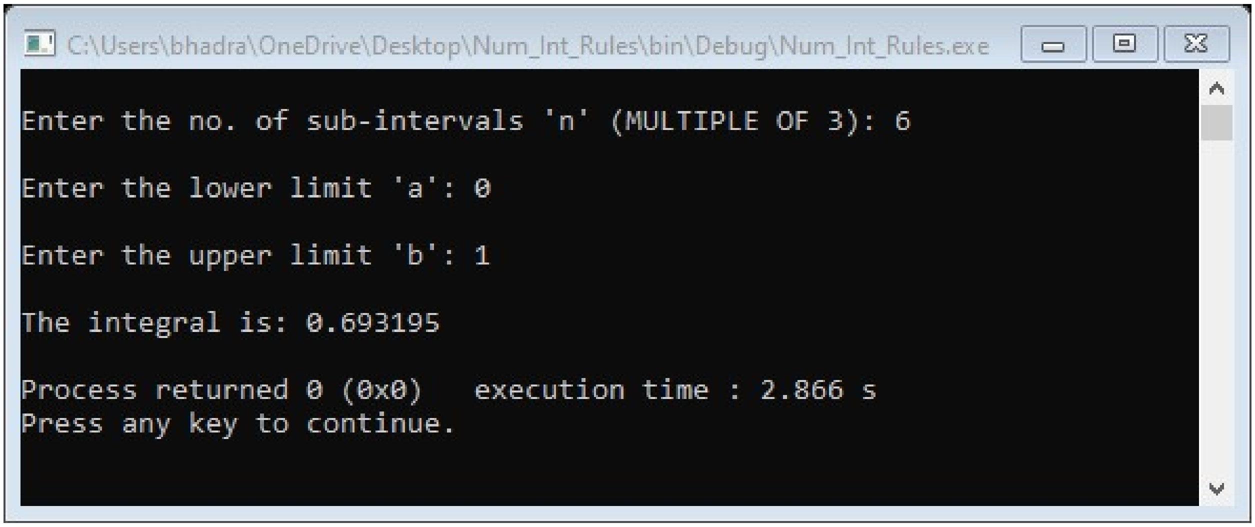
```
      else
       {
         sum=sum+3*f(x);
       }
      }
   integral=(3*h/8)*(f(a)+f(b)+sum);

   printf("\nThe integral is: %lf\n",integral);

  }

else
  {
     printf("Oops!\n    Simpons    three-eighth    rule    is    not
applicable.\n Please enter a number which is a multiple of 3.\n");
   }
}
```

================================ END OF THE PROGRAM ================================

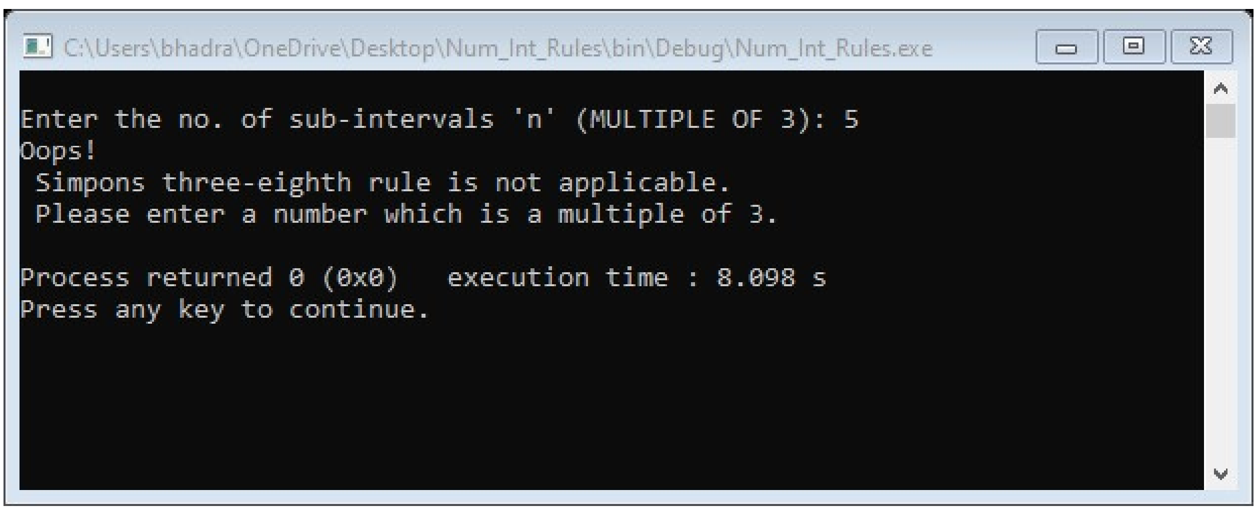
The screeshot of the c program for Simpson’s one-third rule:



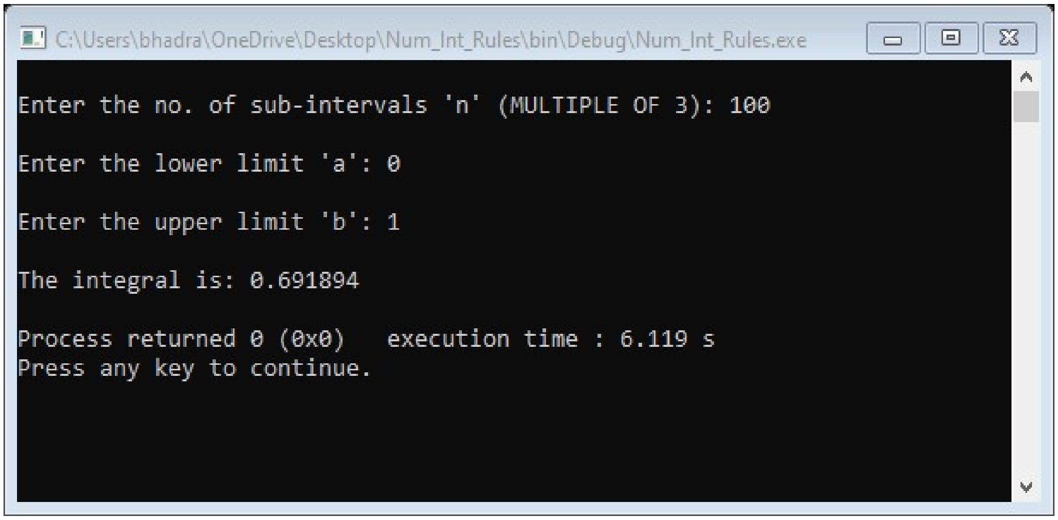
The integral value with n=6:

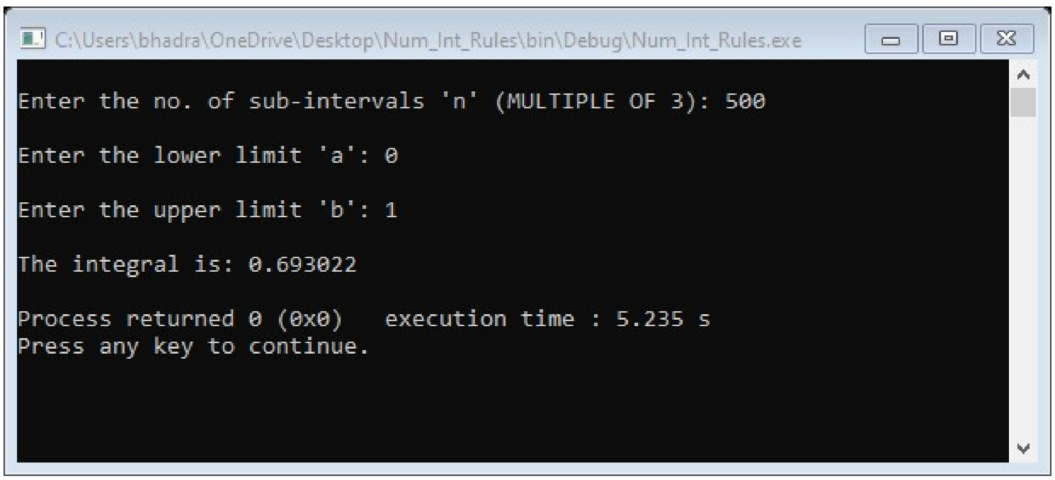


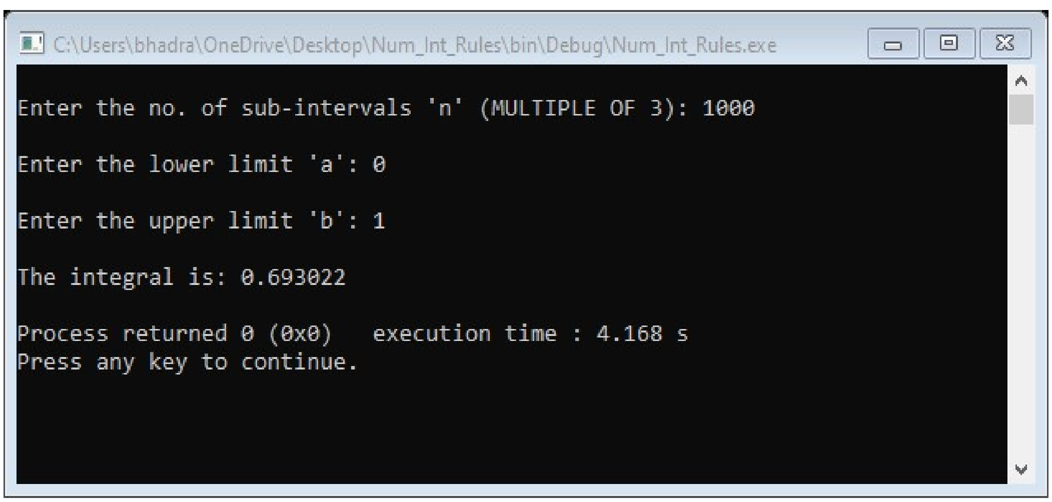
Showing Simpson's three-fourth rule is not applicabe if n is not a multiple of 3:



From the below output screenshots, we can conclude that the exact value of the integral is 0.693022 which is obtained by increasing the number of sub-intervals, but is not matching with the analytical solution. Thus Simpson's one-third rule is more accurate than the three-fourth solution.


C:\Users\bhadra\OneDrive\Desktop\Num_Int_Rules\bin\Debug\Num_Int_Rules.exe
Enter the no. of sub-intervals 'n' (MULTIPLE OF 3): 100
Enter the lower limit 'a': 0
Enter the upper limit 'b': 1
The integral is: 0.691894
Process returned 0 (0x0)   execution time : 6.119 s
Press any key to continue.


C:\Users\bhadra\OneDrive\Desktop\Num_Int_Rules\bin\Debug\Num_Int_Rules.exe
Enter the no. of sub-intervals 'n' (MULTIPLE OF 3): 500
Enter the lower limit 'a': 0
Enter the upper limit 'b': 1
The integral is: 0.693022
Process returned 0 (0x0)   execution time : 5.235 s
Press any key to continue.


C:\Users\bhadra\OneDrive\Desktop\Num_Int_Rules\bin\Debug\Num_Int_Rules.exe
Enter the no. of sub-intervals 'n' (MULTIPLE OF 3): 1000
Enter the lower limit 'a': 0
Enter the upper limit 'b': 1
The integral is: 0.693022
Process returned 0 (0x0)   execution time : 4.168 s
Press any key to continue.

CHAPTER 4

BOOLE'S RULE AND WEDDLE'S RULE WITH C - PROGRAMS

BOOLE'S RULE

Taking $n = 4$ in the the General Quadrature formula, all differences higher than the fourth order will become zero and thus we get

$$\int_{x_0}^{x_n} f(x)dx = \frac{2h}{45}[(7y_0 + 32y_1 + 12y_2 + 32y_3 + 7y_4) + (7y_4 + 32y_5 + 12y_6 + 32y_7 + 7y_8) + \cdots + (7y_{n-4} + 32y_{n-3} + 12y_{n-2} + 32y_{n-1} + 7y_n)]$$

This is known as $Boole's\ rule$. The number of intervals in this rule must be a multiple of 4.

Example Evaluate $\int_0^1 \frac{1}{1+x} dx$ by using $Boole's\ rule$ with four sub-intervals.

Solution Divide the interval $[0,1]$ into four subintervals.

Here $h = \frac{b-a}{4} = \frac{1-0}{4} = \frac{1}{4}$

The values of x and y are tabulated as below:

x	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1
$y = \frac{1}{1+x}$	1	0.8	0.666667	0.5714286	0.5

$\therefore y_0 = 1, y_1 = 0.8, y_2 = 0.666667, y_3 = 0.5714286, y_4 = 0.5$

By Boole's rule,

$$\int_0^1 \frac{1}{1+x} dx = \frac{2h}{45}[7y_0 + 32y_1 + 12y_2 + 32y_3 + 7y_4]$$

$$= \frac{1}{90}[7(1) + 32(0.8) + 12(0.666667) + 32(0.5714286) + 7(0.5)]$$

$$= 0.69317466$$

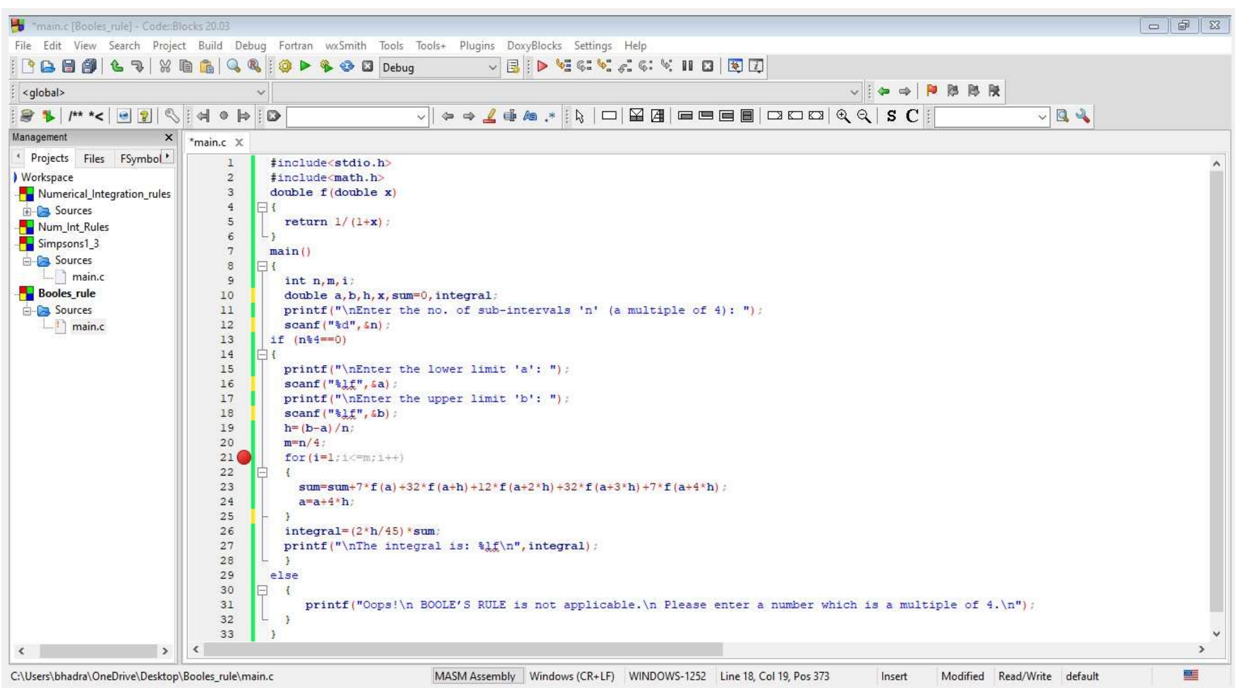
BOOLE'S RULE C-PROGRAM

```
#include<stdio.h>
#include<math.h>

double f(double x)
{
  return 1/(1+x);
}

main()
{
  int n,m,i;
  double a,b,h,x,sum=0,integral;

  printf("\nEnter the no. of sub-intervals 'n' (EVEN): ");
  scanf("%d",&n);

if (n%4==0)
{
  printf("\nEnter the lower limit 'a': ");
  scanf("%lf",&a);

  printf("\nEnter the upper limit 'b': ");
  scanf("%lf",&b);

  h=(b-a)/n;
  m=n/4;

  for(i=1;i<=m;i++)
  {
    sum=sum+7*f(a)+32*f(a+h)+12*f(a+2*h)+32*f(a+3*h)+7*f(a+4*h);
    a=a+4*h;
  }
  integral=(2*h/45)*sum;
```

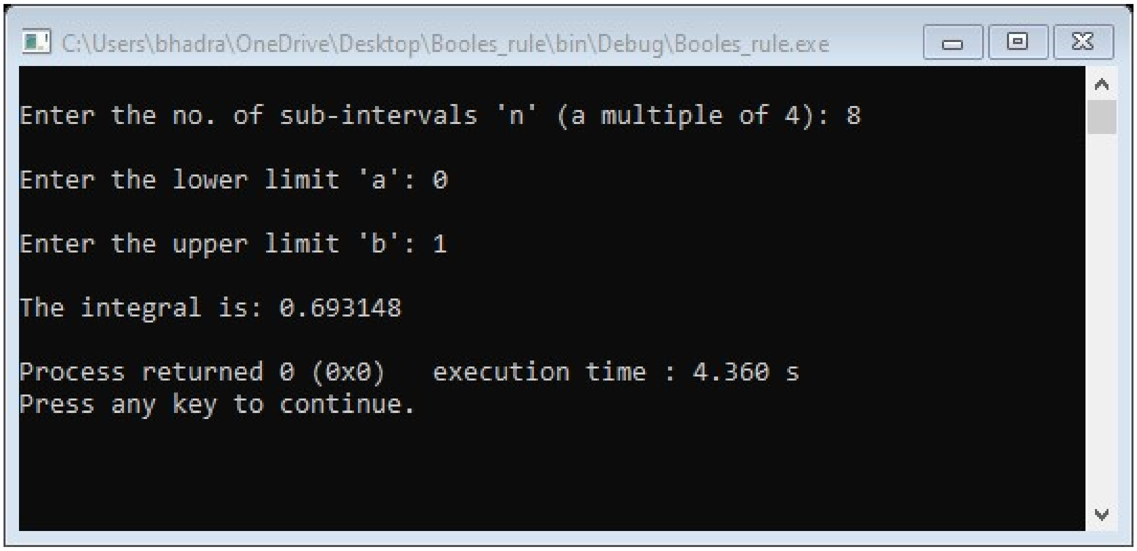
```
  printf("\nThe integral is: %lf\n",integral);

 }

else
  {
     printf("Oops!\n BOOLE'S RULE is not applicable.\n Please
enter a number which is a multiple of 4.\n");
   }
}
```

```
====================== END OF THE PROGRAM ========================
```

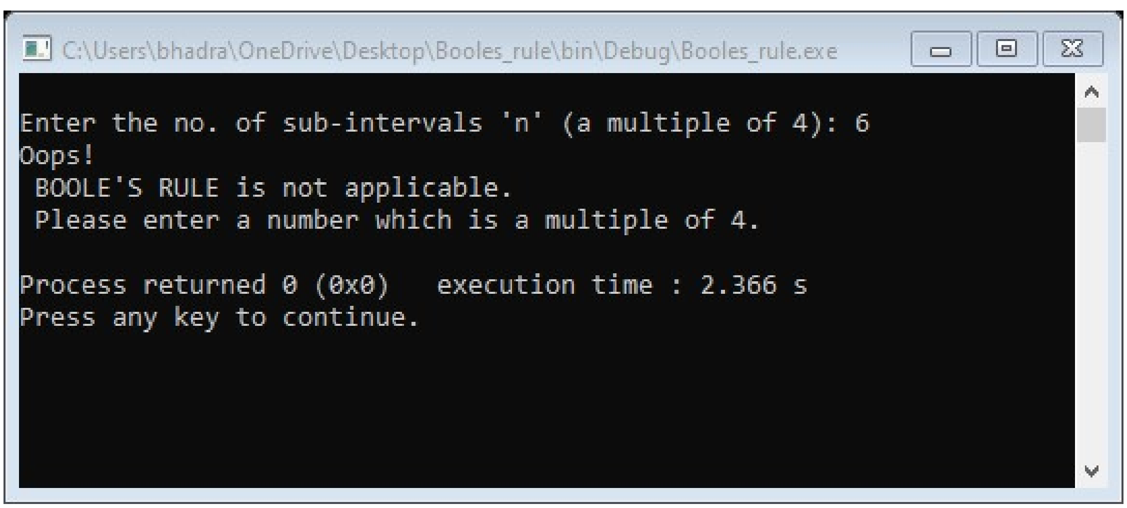
Screenshot of the c-program in CodeBlocks:



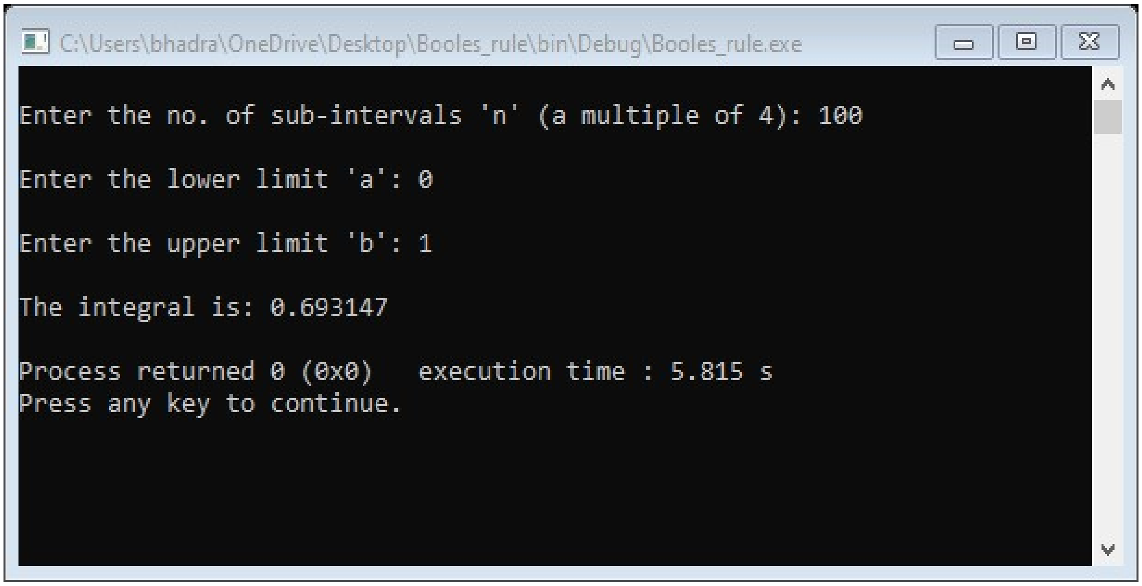
Screenshot of the output showing integral value 0.693148 with $n = 8$:

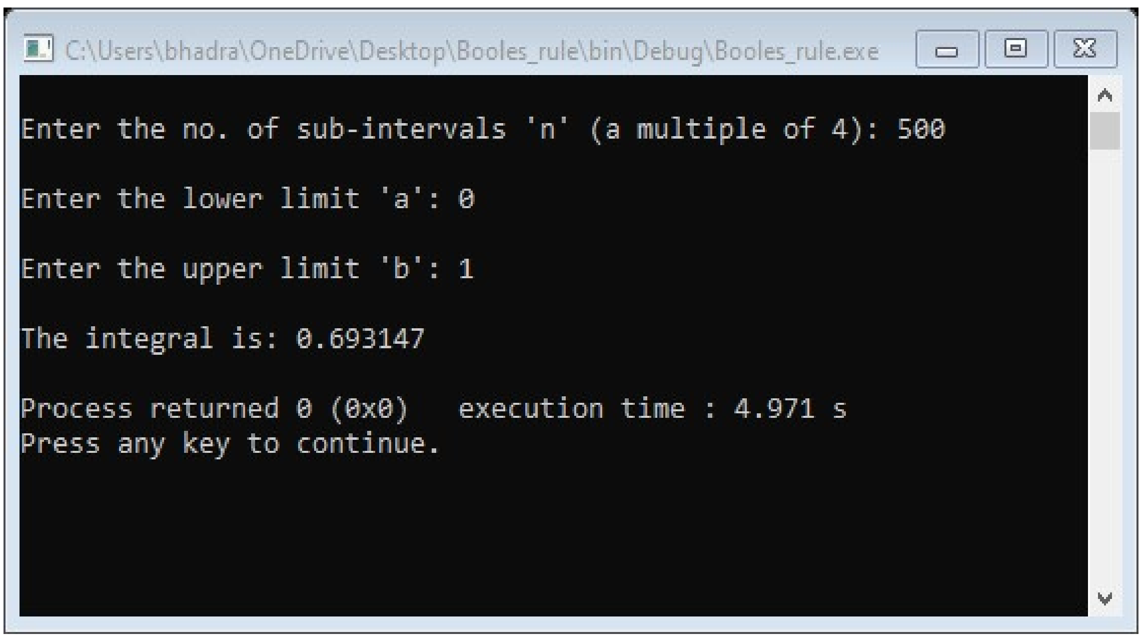


Screenshot showing the integral is not applicable if n is not a multiple of 4:



Screenshots of the output showing integral value 0.693417 which matches with the analytical solution with $n = 100$:





WEDDLE'S RULE

Taking $n = 6$ in the the General Quadrature formula, all differences higher than the sixth order will become zero and thus we get

$$\int_{x_0}^{x_n} f(x)dx = \frac{3h}{10}[(y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6)$$
$$+ (y_6 + 5y_7 + y_8 + 6y_9 + y_{10} + 5y_{11} + y_{12}) + \cdots$$
$$+ (y_{n-6} + 5y_{n-5} + y_{n-4} + 6y_{n-3} + y_{n-2} + 5y_{n-1} + y_n)]$$

This is known as $Weddle's\ rule$. The number of intervals in this rule must be a multiple of 6.

Example Evaluate $\int_0^1 \frac{1}{1+x}dx$ by using $Boole's\ rule$ with six sub-intervals.

Solution Divide the interval $[0,1]$ into six subintervals.

Here $h = \frac{b-a}{6} = \frac{1-0}{6} = \frac{1}{6}$

The values of x and y are tabulated as below:

x	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1
$y = \frac{1}{1+x}$	1	0.857142	0.75	0.666667	0.6	0.545454	0.5

$\therefore y_0 = 1, y_1 = 0.857142, y_2 = 0.75, y_3 = 0.6667, y_4 = 0.6, y_5 = 0.545454, y_6 = 0.5$

By Weddle's rule,

$$\int_0^1 \frac{1}{1+x}dx = \frac{3h}{10}[y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6]$$

$$= \frac{1}{20}[1 + 5(0.857142 + 0.75 + 6(0.666667) + 0.6 + 5(0.545454 + 0.5)]$$

$$= 0.693149$$

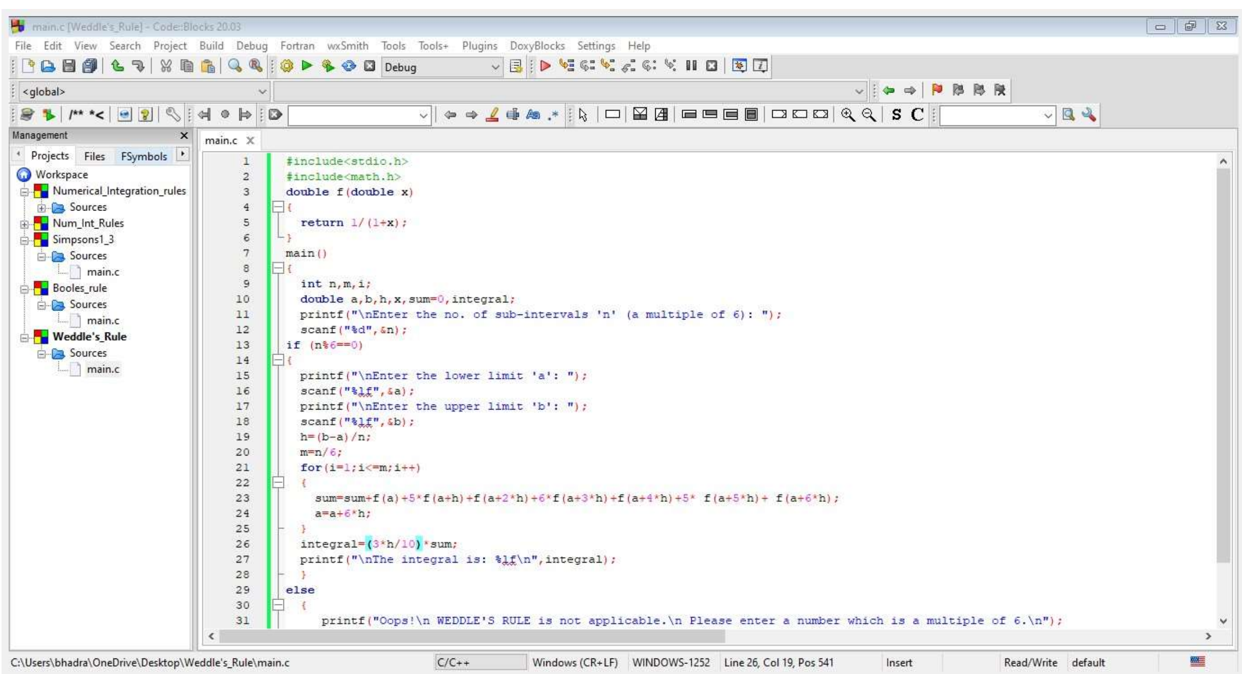
WEDDLE'S RULE C-PROGRAM

```
#include<stdio.h>
#include<math.h>

double f(double x)
{
  return 1/(1+x);
}

main()
{
  int n,m,i;
  double a,b,h,x,sum=0,integral;

  printf("\nEnter the no. of sub-intervals 'n' (EVEN): ");
  scanf("%d",&n);

if (n%6==0)
{
  printf("\nEnter the lower limit 'a': ");
  scanf("%lf",&a);

  printf("\nEnter the upper limit 'b': ");
  scanf("%lf",&b);

  h=(b-a)/n;
  m=n/6;

  for(i=1;i<=m;i++)
  {
    sum=sum+f(a)+5*f(a+h)+f(a+2*h)+6*f(a+3*h)+f(a+4*h)+5*
f(a+5*h)+ f(a+6*h);
    a=a+6*h;
  }
```

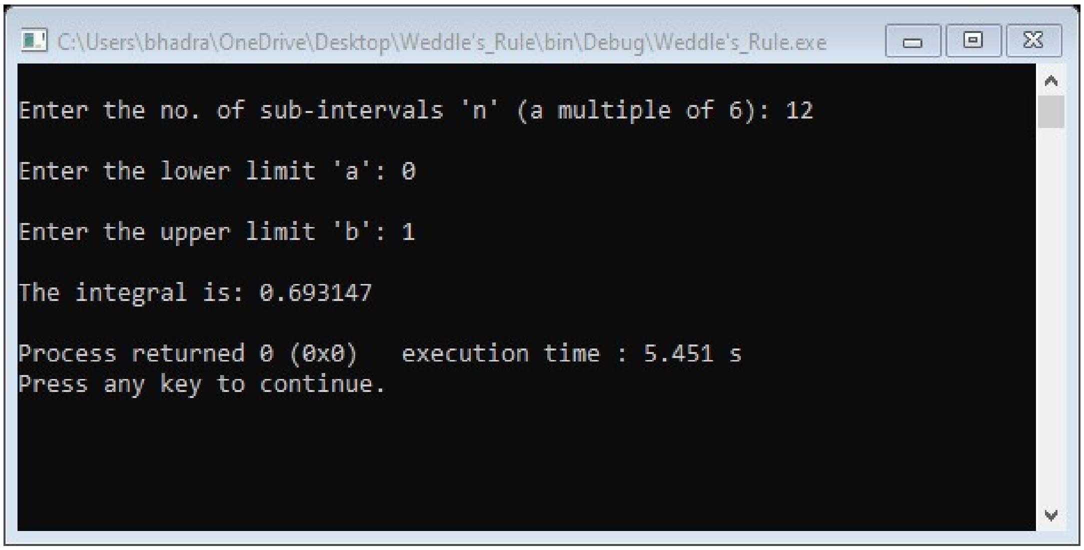
```
    integral=(3*h/10)*sum;

    printf("\nThe integral is: %lf\n",integral);

  }

else
   {
      printf("Oops!\n WEDDLE’S RULE is not applicable.\n Please
enter a number which is a multiple of 6.\n");
    }
}
```

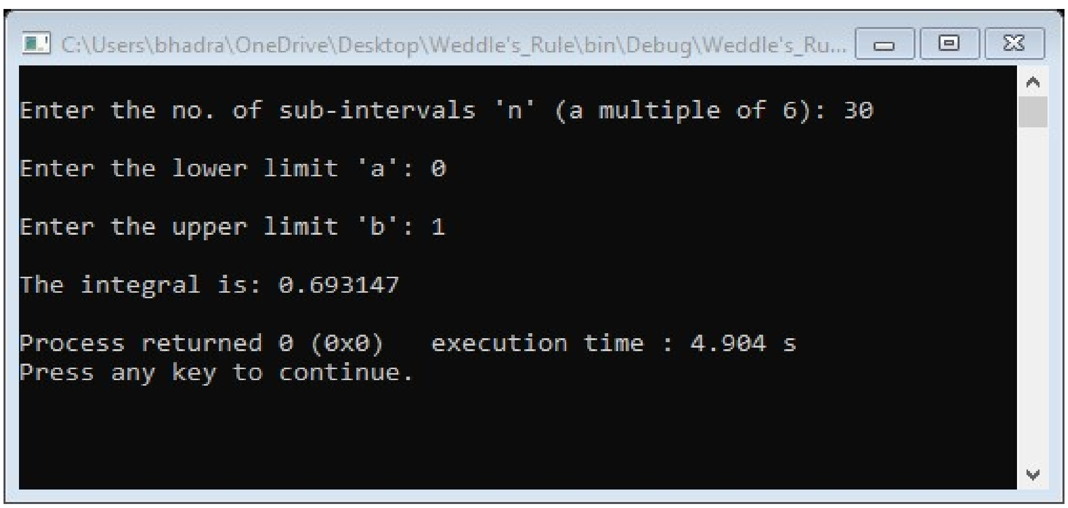
====================== END OF THE PROGRAM ======================

Screenshot of the c-program:

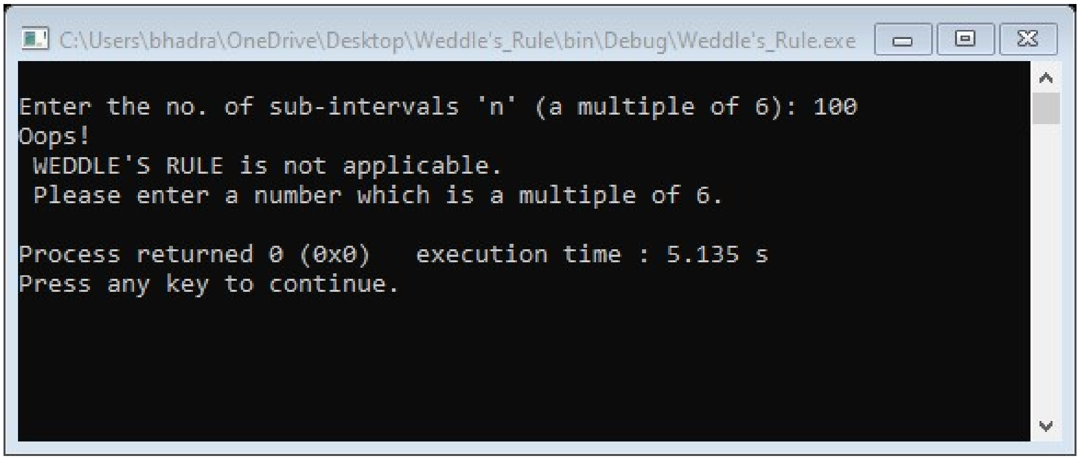


Screenshot of the output showing the integral value 0.693147 which matches with the analytical solution with just $n = 6$:





Screenshot of the output showing that the rule is not applicable if n is not a multiple of 6:



CONCLUSION

The value of the integral in Trapezoidal rule matches with the analytical solution with the n value near to 500 and in case of , Simpson's one-third rule and in Boole's rule it is obtained with with the n value near to 100 and with the Weddle's rule it is obtained with just $n = 12$. Thus Simpson's one-third rule is more accurate than the Trapezoidal rule.

Among all the rules, Weddle's Rule is more accurate than any of the other rules.

References

1. C-Language Tutorial videos by Srinivas, Naresh Technologies

   https://youtube.com/playlist?list=PLVlQHNRLflP8IGz6OXwlV_lgHgc72aXlh

2. Finite Differences and Numerical Analysis by H.C Saxena published by S.Chand and Company, Pvt. Ltd., New Delhi.

3. Numercial Analysis by S.Ranganatham, MVSSN Prasad and V.Ramesh Babu, S.Chand Publications.