Assignment - Numerical Integration

Rajan Maurya (72004) *V.1 - Algorithms for Computational Mathematics: Numerical Methods*

B. Tech. (Information Technology and Mathematical Innovations)

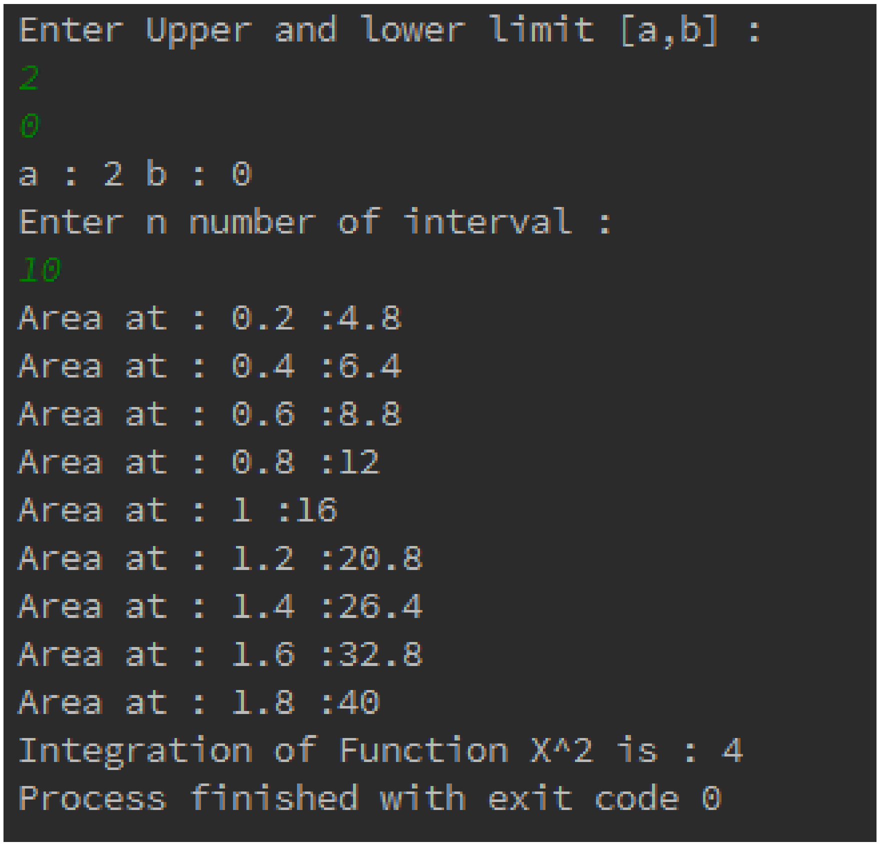
Cluster Innovation Centre, University of Delhi

Trapezoidal rule

```
#include <iostream>
#include <cmath>
```

using namespace std; /** * Created by rajan on 2/12/15. */

```
/**
 *  Function to integrate
 */
float Ifunction(float x)
{

    return (2*x);
}


/**
 * a is Upper limit
 * b is lower limit
 * n is number of interval
 * h is interval gap
 * sum is area under the curve
 */
float x,a,b,n,h,sum = 0;

int main()
{
    cout<<"Enter Upper and lower limit [a,b] :"<<endl;
    cin>>a>>b;
    cout<<"a : "<<a<<" b : "<<b<<endl;
    cout<<"Enter n number of interval : "<<endl;
    cin>>n;

    h = (a-b)/n;
    x = h;
    sum = Ifunction(a) + Ifunction(b);

    while (x <a)
    {

        sum += 2*Ifunction(x);
        cout<<"Area at : "<<x<<" :"<< sum << endl;
        x+=h;

    }

    cout<<"Integration of Function 2*X is : " << (h/2)*sum;

}
```



```
Enter Upper and lower limit [a,b] :
2
0
a : 2 b : 0
Enter n number of interval :
10
Area at : 0.2 :4.8
Area at : 0.4 :6.4
Area at : 0.6 :8.8
Area at : 0.8 :12
Area at : 1 :16
Area at : 1.2 :20.8
Area at : 1.4 :26.4
Area at : 1.6 :32.8
Area at : 1.8 :40
Integration of Function X^2 is : 4
Process finished with exit code 0
```

Figure 1: Bisection Method

Output

Simpson's rule

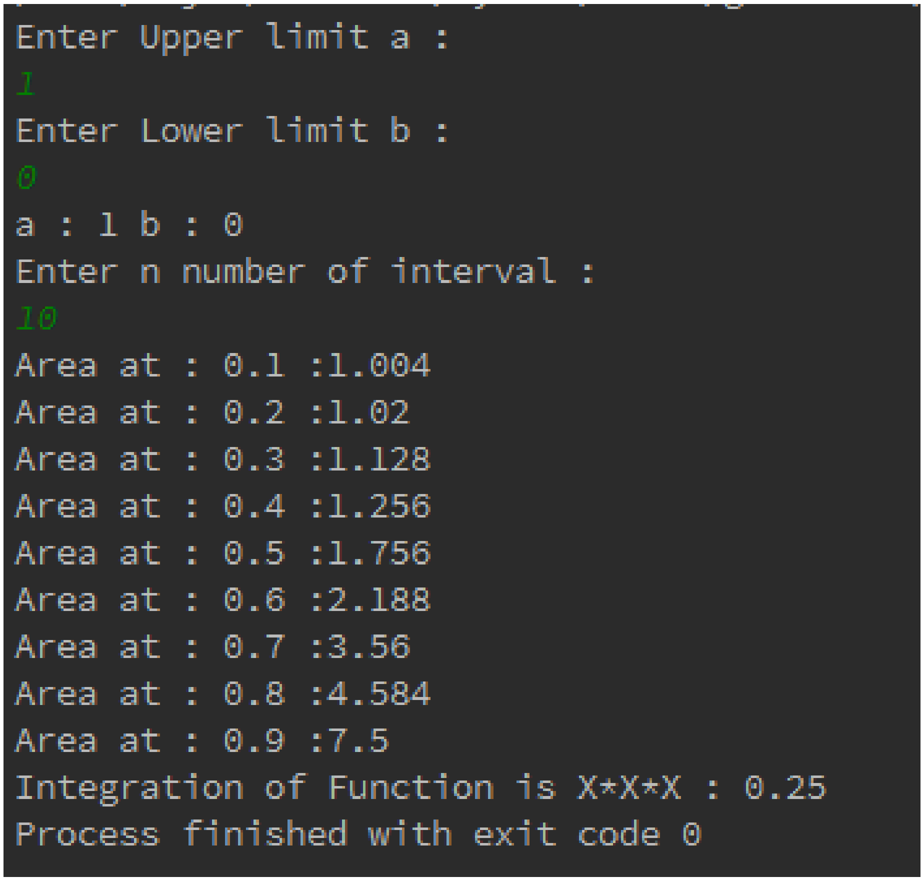
```
#include <iostream>
#include <cmath>

using namespace std;
/**
 - Created by rajan on 2/12/15.
 */

/**
 *  Function to integrate
 */
float Ifunction(float x)
{

    return (x*x*x);
}


/**
 * a is Upper limit
 * b is lower limit
 * n is number of interval
 * h is interval gap
 * sum is area under the curve
 */
float x,a,b,n,h,sum = 0;

int main()
{
    cout<<"Enter Upper limit a :"<<endl;
    cin>>a;
    cout<<"Enter Lower limit b :"<<endl;
    cin>>b;
    cout<<"a : "<<a<<" b : "<<b<<endl;
    cout<<"Enter n number of interval : "<<endl;
    cin>>n;

    h = (a-b)/n;

    x = h;
    sum = Ifunction(a) + Ifunction(b);
```

```
   for (int i=1 ; i<n ; i++)
    {
        if(i%2 == 0)
        {

            sum += 2*Ifunction(x);

        } else
        {

            sum += 4*Ifunction(x);

        }
        cout<<"Area at : "<<x<<" :"<< sum << endl;
        x+=h;
    }

    cout<<"Integration of Function is X*X*X : " << (h/3)*sum;

}
```

Output

Gaussian_one_point_1D

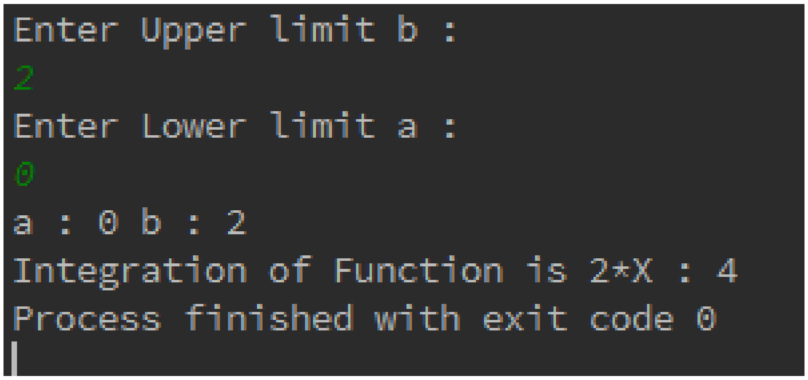
```
#include <iostream>
#include <cmath>

using namespace std;
/**
 - Created by rajan on 2/12/15.
 */

/**
 -  Function to integrate
 */
float Ifunction(float x)
{

    return (2*x);
}
```



```
Enter Upper limit a :
1
Enter Lower limit b :
0
a : 1 b : 0
Enter n number of interval :
10
Area at : 0.1 :1.004
Area at : 0.2 :1.02
Area at : 0.3 :1.128
Area at : 0.4 :1.256
Area at : 0.5 :1.756
Area at : 0.6 :2.188
Area at : 0.7 :3.56
Area at : 0.8 :4.584
Area at : 0.9 :7.5
Integration of Function is X*X*X : 0.25
Process finished with exit code 0
```

Figure 2: Simpson'Rule

```
/**
     - a is Upper limit
     - b is lower limit
     - n is number of interval for two point n=1
     */
float w = 1 ,a,b,n = 1;

int main()
{

cout<<"Enter Upper limit b :"<<endl;
cin>>b;
cout<<"Enter Lower limit a :"<<endl;
cin>>a;
cout<<"a : "<<a<<" b : "<<b<<endl;

double I = ((b-a)*Ifunction((a+b)/2));

cout<<"Integration of Function is X^2 : " << I;

}
```



```
Enter Upper limit b :
2
Enter Lower limit a :
0
a : 0 b : 2
Integration of Function is 2*X : 4
Process finished with exit code 0
```

Figure 3: Gaussian_one_point_1D Method

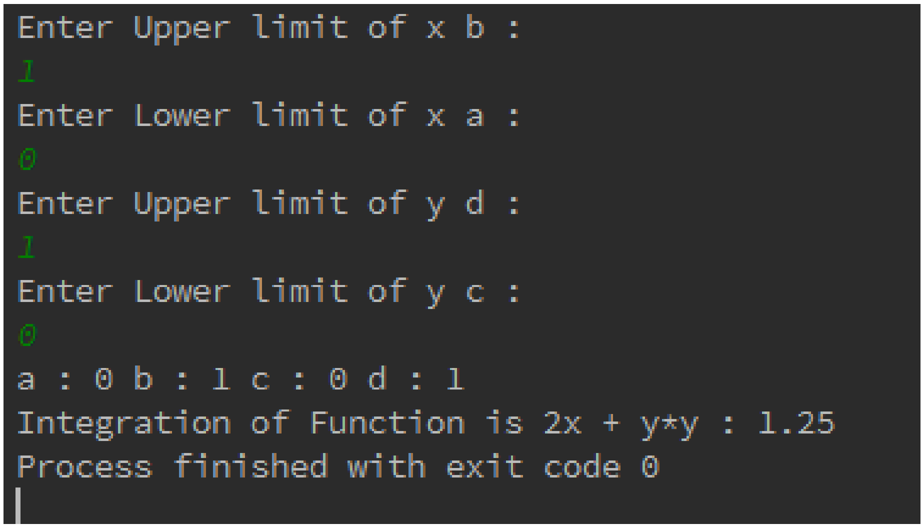
Output

Gauss_one_point_2D

```
#include <iostream>
#include <cmath>
```

```
using namespace std;
/**
 - Created by rajan on 2/12/15.
 */

/**
 -  Function to integrate
 */
double Ifunction(double x , double y)
{

return (2*x + y*y);


}


/**
     - a is Upper limit
     - b is lower limit
     - n is number of interval for two point n=1
     */
float w = 1 ,a,b,c,d,n = 1;

int main()
{

cout<<"Enter Upper limit of x b :"<<endl;
cin>>b;
cout<<"Enter Lower limit of x a :"<<endl;
cin>>a;
cout<<"Enter Upper limit of y d :"<<endl;
cin>>d;
cout<<"Enter Lower limit of y c :"<<endl;
cin>>c;
cout<<"a : "<<a<<" b : "<<b << " c : " <<c << " d : " <<d<<endl;

double c1 = (a+b)/2;
double c2 = (c+d)/2;

double I = ((b-a)*(d-c)*Ifunction(c1,c2));
cout<<"Integration of Function is 2x + y*y : " << I;

}
```

Output



```
Enter Upper limit of x b :
1
Enter Lower limit of x a :
0
Enter Upper limit of y d :
1
Enter Lower limit of y c :
0
a : 0 b : 1 c : 0 d : 1
Integration of Function is 2x + y*y : 1.25
Process finished with exit code 0
```

Figure 4: G12D

Gauss Two point_1D

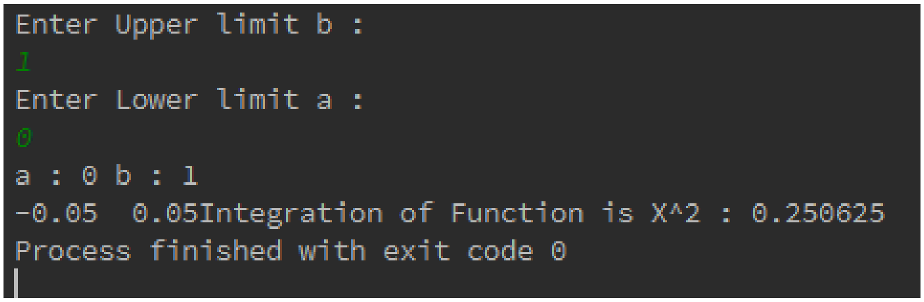
```
   #include <iostream>
   #include <cmath>

using namespace std;
/**
 * Created by rajan on 2/12/15.
 */

/**
 *  Function to integrate
 */
float Ifunction(float x)
{

    return (x*x);
}

float Tx(float x1 , float a , float b)
{

    return ((1-x1)*a + (1+x1)*b)/2;
}
```

```
/**
     * a is Upper limit
     * b is lower limit
     * n is number of interval for two point n=2
     * h is interval gap
     * w is the weight w1 w2 both one so assume w as single
     * sum is area under the curve
     */
float w = 1 ,a,b,n = 2,h,sum = 0 ;
double x1 = -0.05 , x2 = 0.05;

int main()
{

    cout<<"Enter Upper limit b :"<<endl;
    cin>>b;
    cout<<"Enter Lower limit a :"<<endl;
    cin>>a;
    cout<<"a : "<<a<<" b : "<<b<<endl;

    cout<<x1<<"  "<<x2;
    float tx1 = Tx(x1 , a , b);
    float tx2 = Tx(x2 , a , b);

    float Ix = w*(Ifunction(tx1) + Ifunction(tx2));

    //final Integration is I

    float I = ((b-a)*Ix)/2;


    cout<<"Integration of Function is X^2 : " << I;

}
```

Output

Gauss Two point_2D

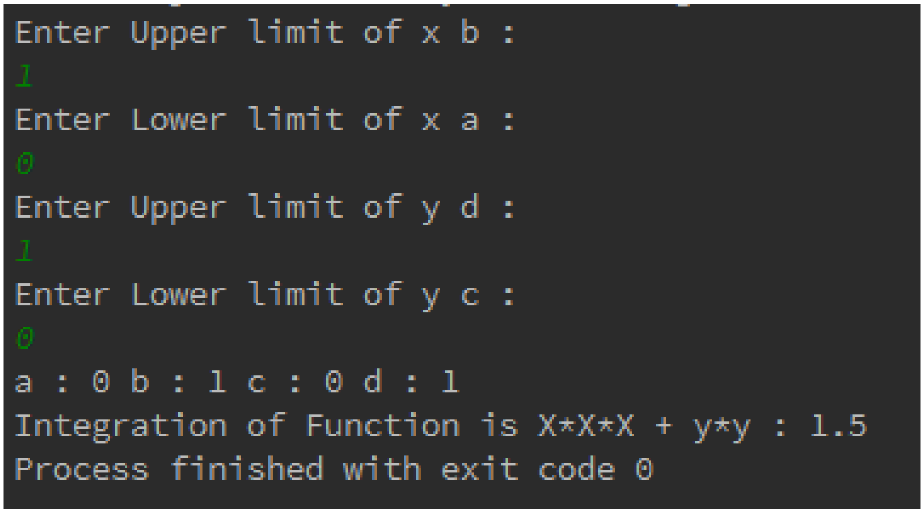
```
#include <iostream>
#include <cmath>

   using namespace std;
   /**
```



```
Enter Upper limit b :
1
Enter Lower limit a :
0
a : 0 b : 1
-0.05  0.05Integration of Function is X^2 : 0.250625
Process finished with exit code 0
```

Figure 5: G21D

```
     + Created by rajan on 2/12/15.
     */

/**
 *  Function to integrate
 */
double Ifunction(double a , double b , double c , double d , double t1 , double t2)
{

    double x = (a+b)/2 + ((b-a)/2)*t1;
    double y = (c+d)/2 + ((d-c)/2)*t2;

    return (x*x*x + y*y);
}


/**
     * a is Upper limit
     * b is lower limit
     * n is number of interval for two point n=1
     */
double  a,b,c,d,n = 1 , result = 0;
double t11= -pow(1/3,0.5) , t12 = pow(1/3,0.5) , t21 = -pow(1/3,0.5) , t22 =  pow(1/3,0.5);

int main()
{

    cout<<"Enter Upper limit of x b :"<<endl;
    cin>>b;
    cout<<"Enter Lower limit of x a :"<<endl;
    cin>>a;
    cout<<"Enter Upper limit of y d :"<<endl;
    cin>>d;
    cout<<"Enter Lower limit of y c :"<<endl;
    cin>>c;
    cout<<"a : "<<a<<" b : "<<b << " c : " <<c << " d : " <<d<<endl;

    result = result + Ifunction(a,b,c,d,t11 ,t21) + Ifunction(a,b,c,d,t11 ,t22) + Ifunction(
            +Ifunction(a,b,c,d,t12 ,t22);

    cout<<"Integration of Function is X*X*X + y*y : " << result;

}
```



```
Enter Upper limit of x b :
1
Enter Lower limit of x a :
0
Enter Upper limit of y d :
1
Enter Lower limit of y c :
0
a : 0 b : 1 c : 0 d : 1
Integration of Function is X*X*X + y*y : 1.5
Process finished with exit code 0
```

Figure 6: G22d

Output **THE END.**