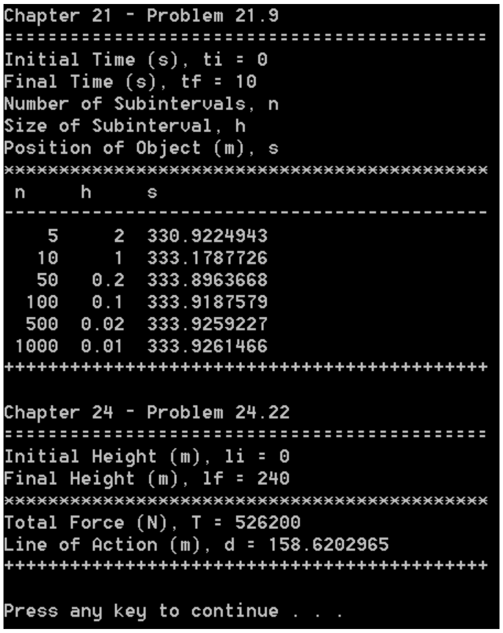
Problems 1 & 2 – Trapezoidal Rule (Multiple-Segment)

```cpp
// 03/25/2014 - ENGR 2450 - Meine, Joel
// Problems 21.9, 24.22

// Trapezoidal Rule (Multiple-Segment)

#include <iostream>
#include <iomanip>
#include <math.h>
#include <vector>
using namespace std;

double Trapm(double h,int n,vector<double> f)
{
        double sum = f[0];
        for (int i = 1; i < n; i++)
                sum = sum + 2 * f[i];
        sum = sum + f[n];
        return(h * (sum/2));
}

double V(double t)
{
        const double g = 9.8; // Gravitational Constant (m/s^2), g
        const double m = 68.1; // Mass of Object (kg), m
        const double cd = 0.25; // Drag Coefficient (kg/m), cd
        double v = sqrt((g*m)/cd)*tanh(sqrt((g*cd)/m)*t); //
Velocity of Object (m/s), v
        return(v);
}

void funcEval(double a,double b,int n,double& h,double& I)
{
        h = (b-a)/n;
        vector<double> y;
        for (int i = 0; i <= n; i++)
        {
                y.push_back(V(a+i*h));
        }
        I = Trapm(h,n,y);
        return;
}

void dataEval(double a,double b,int n,vector<double> y,double& I)
{
        double h = (b-a)/n;
        I = Trapm(h,n,y);
        return;
}

int main()
{
        double h = 0; // Size of Subinterval, h

        // Problem 21.9 - Integral of a Function
```



```
Chapter 21 - Problem 21.9
============================================
Initial Time (s), ti = 0
Final Time (s), tf = 10
Number of Subintervals, n
Size of Subinterval, h
Position of Object (m), s
********************************************
 n     h      s
--------------------------------------------
    5     2  330.9224943
   10     1  333.1787726
   50   0.2  333.8963668
  100   0.1  333.9187579
  500  0.02  333.9259227
 1000  0.01  333.9261466
++++++++++++++++++++++++++++++++++++++++++++

Chapter 24 - Problem 24.22
============================================
Initial Height (m), li = 0
Final Height (m), lf = 240
********************************************
Total Force (N), T = 526200
Line of Action (m), d = 158.6202965
++++++++++++++++++++++++++++++++++++++++++++

Press any key to continue . . .
```

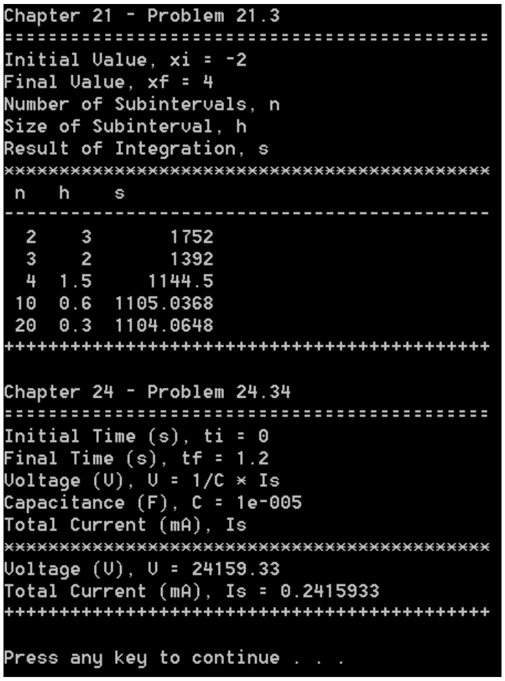
$$s(t)=\int_{0}^{10} v(t)\,dt = 333.9262$$

$$v(t)=\sqrt{\frac{gm}{c_d}}\tanh\left(\sqrt{\frac{gc_d}{m}}t\right)$$

```
       const double ti = 0; // Initial Time (s), t
       const double tf = 10; // Final Time (s), t
       double s = 0; // Position of Object (m), s

       std::cout << "Chapter 21 - Problem 21.9" << std::endl;
       std::cout << "============================================" << std::endl;
       std::cout << "Initial Time (s), ti = " << setprecision(0) << ti << std::endl;
       std::cout << "Final Time (s), tf = " << setprecision(0) << tf << std::endl;
       std::cout << "Number of Subintervals, n" << std::endl;
       std::cout << "Size of Subinterval, h" << std::endl;
       std::cout << "Position of Object (m), s" << std::endl;
       std::cout << "********************************************" << std::endl;
       std::cout << " n     h     s" << std::endl;
       std::cout << "--------------------------------------------" << std::endl;
       const int n_size = 6;
       int n[] = {5,10,50,100,500,1000};
       for (int i = 0; i < n_size; i++)
       {
              funcEval(ti,tf,n[i],h,s);
              cout << setw(5) << n[i];
              cout << setw(6) << setprecision(3) << h;
              cout << setw(13) << setprecision(10) << s << endl;
       }
       std::cout << "++++++++++++++++++++++++++++++++++++++++++++" << std::endl;
       cout << "\n";

       // Problem 24.22 - Integral for Tabulated Data
       vector<double> l = {0,30,60,90,120,150,180,210,240}; // Height (m), l
       vector<double> F = {0,340,1200,1600,2700,3100,3200,3500,3800}; // Force (N/m), F(l)
       vector<double> lF;
       for (int i = 0; i < l.size(); i++)
              lF.push_back(l[i]*F[i]);
       double li = l.front(); // Initial Height (m), li
       double lf = l.back(); // Final Height (m), lf
       double T = 0; // Total Force (N), T
       double d = 0; // Line of Action (m), d

       std::cout << "Chapter 24 - Problem 24.22" << std::endl;
       std::cout << "============================================" << std::endl;
       std::cout << "Initial Height (m), li = " << setprecision(0) << li << std::endl;
       std::cout << "Final Height (m), lf = " << setprecision(0) << lf << std::endl;
       std::cout << "********************************************" << std::endl;

       dataEval(li,lf,F.size()-1,F,T);
       std::cout << "Total Force (N), T = " << setprecision(10) << T << std::endl;

       double dT = 0;
       dataEval(li,lf,lF.size()-1,lF,dT);
       d = dT/T;
       std::cout << "Line of Action (m), d = " << setprecision(10) << d << std::endl;
       std::cout << "++++++++++++++++++++++++++++++++++++++++++++" << std::endl;
       cout << "\n";

       system("pause");
       return 0;
}
```

Problems 3 & 4 – Simpson’s Rules (Equally-Spaced)

```
// 03/25/2014 - ENGR 2450 - Meine, Joel
// Problems 21.3, 24.34

// Simpson's Rules (Equally-Spaced)

#include <iostream>
#include <iomanip>
#include <math.h>
#include <vector>
using namespace std;

double Trap(double h,double f0,double f1)
{
        return((h/2) * (f0+f1));
}

double Simp38(double h,double f0,double f1,double f2,double f3)
{
        return(3*h * ((f0+3*(f1+f2)+f3)/8));
}

double Simp13m(double h,int n,vector<double> f)
{
        double sum = f[0];
        for (int i = 1; i < (n-2); i = i+2)
                sum = sum + 4*f[i] + 2*f[i+1];
        sum = sum + 4*f[n-1] + f[n];
        return(h * (sum/3));
}

double SimpInt(double h,double a,double b,int n,vector<double> f)
{
        double sum = 0;
        if (n == 1)
                sum = Trap(h,f[n-1],f[n]);
        else
        {
                int m = n;
                int odd = n%2;
                if (odd == 1 && n > 1)
                {
                        sum = sum + Simp38(h,f[n-3],f[n-2],f[n-1],f[n]);
                        m = n - 3;
                }
                if (m > 1)
                        sum = sum + Simp13m(h,m,f);

        }
        return(sum);
}

double F(double x)
{
        return(1 - x - 4*pow(x,3) + 2*pow(x,5));
```



```
Chapter 21 - Problem 21.3
===========================================
Initial Value, xi = -2
Final Value, xf = 4
Number of Subintervals, n
Size of Subinterval, h
Result of Integration, s
*******************************************
 n   h    s
-------------------------------------------
  2    3       1752
  3    2       1392
  4  1.5     1144.5
 10  0.6  1105.0368
 20  0.3  1104.0648
+++++++++++++++++++++++++++++++++++++++++++

Chapter 24 - Problem 24.34
===========================================
Initial Time (s), ti = 0
Final Time (s), tf = 1.2
Voltage (V), V = 1/C * Is
Capacitance (F), C = 1e-005
Total Current (mA), Is
*******************************************
Voltage (V), V = 24159.33
Total Current (mA), Is = 0.2415933
+++++++++++++++++++++++++++++++++++++++++++

Press any key to continue . . .
```

$$s(x)=\int_{-2}^{4} f(x)dx=1104$$

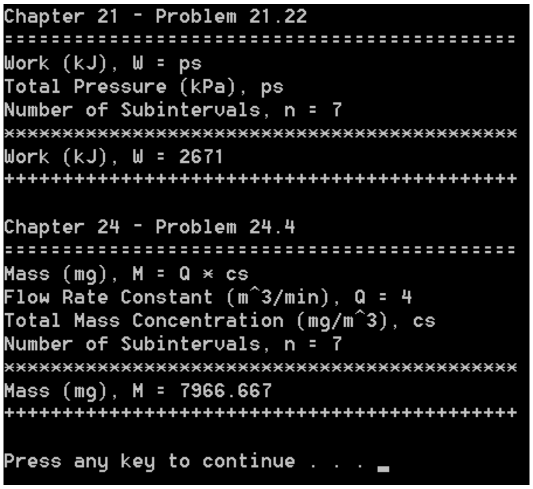
$$f(x)=1-x-4x^3+2x^5$$

```
}

void funcEval(double a,double b,int n,double& h,double& I)
{
       h = (b-a)/n;
       vector<double> y;
       for (int i = 0; i <= n; i++)
       {
              y.push_back(F(a+i*h));
       }
       I = SimpInt(h,a,b,n,y);
       return;
}

void dataEval(double a,double b,int n,vector<double> y,double& I)
{
       double h = (b-a)/n;
       I = SimpInt(h,a,b,n,y);
       return;
}

int main()
{
       double h = 0; // Size of Subinterval, h

       // Problem 21.3 - Integral of a Function
       const double xi = -2; // Initial Value, xi
       const double xf = 4; // Final Value, xf
       double s = 0; // Result of Integration, s

       std::cout << "Chapter 21 - Problem 21.3" << std::endl;
       std::cout << "============================================" << std::endl;
       std::cout << "Initial Value, xi = " << setprecision(0) << xi << std::endl;
       std::cout << "Final Value, xf = " << setprecision(0) << xf << std::endl;
       std::cout << "Number of Subintervals, n" << std::endl;
       std::cout << "Size of Subinterval, h" << std::endl;
       std::cout << "Result of Integration, s" << std::endl;
       std::cout << "********************************************" << std::endl;
       std::cout << " n   h    s" << std::endl;
       std::cout << "--------------------------------------------" << std::endl;
       const int n_size = 5;
       int n[] = {2,3,4,10,20};
       for (int i = 0; i < n_size; i++)
       {
              funcEval(xi,xf,n[i],h,s);
              cout << setw(3) << n[i];
              cout << setw(5) << setprecision(3) << h;
              cout << setw(11) << setprecision(9) << s << endl;
       }
       std::cout << "++++++++++++++++++++++++++++++++++++++++++++" << std::endl;
       cout << "\n";

       // Problem 24.34 - Integral for Tabulated Data
       vector<double> t = {0.0,0.2,0.4,0.6,0.8,1.0,1.2}; // Time (s), t
       vector<double> I = {0.2,0.3683,0.3819,0.2282,0.0486,0.0082,0.1441}; // Current (mA), I(t)
       double ti = t.front(); // Initial Time (s), ti
       double tf = t.back(); // Final Time (s), tf
```

```
	const double C = 10 * pow(10,-6); // Capacitance (F), C
	double Is = 0; // Total Current (mA), Is

	std::cout << "Chapter 24 - Problem 24.34" << std::endl;
	std::cout << "============================================" << std::endl;
	std::cout << "Initial Time (s), ti = " << setprecision(0) << ti << std::endl;
	std::cout << "Final Time (s), tf = " << setprecision(0) << tf << std::endl;
	std::cout << "Voltage (V), V = 1/C * Is" << std::endl;
	std::cout << "Capacitance (F), C = " << setprecision(7) << C << std::endl;
	std::cout << "Total Current (mA), Is" << std::endl;
	std::cout << "********************************************" << std::endl;

	dataEval(ti,tf,I.size()-1,I,Is);
	double V = (1/C) * Is; // Voltage (V), V
	std::cout << "Voltage (V), V = " << setprecision(7) << V << std::endl;
	std::cout << "Total Current (mA), Is = " << setprecision(7) << Is << std::endl;

	std::cout << "++++++++++++++++++++++++++++++++++++++++++++" << std::endl;
	cout << "\n";

	system("pause");
	return 0;
}
```

Problems 5 & 6 – Trapezoidal Rule & Simpson's Rules (Unequally-Spaced)

```
// 03/25/2014 - ENGR 2450 - Meine, Joel
// Problems 21.22, 24.4

// Trapezoidal Rule & Simpson's Rules (Unequally-Spaced)

#include <iostream>
#include <iomanip>
#include <math.h>
#include <vector>
using namespace std;

double Trapun(vector<double> x,vector<double> y,int n)
{
	double sum = 0;
	for (int i = 1; i <= n; i++)
		sum = sum + (((x[i]-x[i-1])*(y[i-1]+y[i]))/2);
	return(sum);
}

double Trap(double h,double f0,double f1)
{
	return(h * ((f0+f1)/2));
}

double Simp38(double h,double f0,double f1,double f2,double f3)
{
	return(3*h * ((f0+3*(f1+f2)+f3)/8));
}
```



```
Chapter 21 - Problem 21.22
============================================
Work (kJ), W = ps
Total Pressure (kPa), ps
Number of Subintervals, n = 7
********************************************
Work (kJ), W = 2671
++++++++++++++++++++++++++++++++++++++++++++

Chapter 24 - Problem 24.4
============================================
Mass (mg), M = Q * cs
Flow Rate Constant (m^3/min), Q = 4
Total Mass Concentration (mg/m^3), cs
Number of Subintervals, n = 7
********************************************
Mass (mg), M = 7966.667
++++++++++++++++++++++++++++++++++++++++++++

Press any key to continue . . .
```

```
double Simp13(double h,double f0,double f1,double f2)
{
        return(2*h * ((f0+4*f1+f2)/6));
}

double Uneven(int n,vector<double> x,vector<double> f)
{
        double h = x[1] - x[0];
        int k = 1;
        double sum = 0;
        double hf = 0;
        for (int j = 1; j <= n; j++)
        {
                if (j == n)
                        hf = x[0] - x[j];
                else
                        hf = x[j+1] - x[j];
                if (abs(h-hf) < .000001)
                {
                        if (k == 3)
                        {
                                sum = sum + Simp13(h,f[j-3],f[j-2],f[j-1]);
                                k = k - 1;
                        }
                        else
                                k = k + 1;
                }
                else
                {
                        if (k == 1)
                        {
                                sum = sum + Trap(h,f[j-1],f[j]);
                        }
                        else
                        {
                                if (k == 2)
                                {
                                        sum = sum + Simp13(h,f[j-2],f[j-1],f[j]);
                                }
                                else
                                {
                                        sum = sum + Simp38(h,f[j-3],f[j-2],f[j-1],f[j]);
                                }
                                k = 1;
                        }
                }
                h = hf;
        }
        return(sum);
}

double F(double x)
{
        return(1 - x - 4*pow(x,3) + 2*pow(x,5));
}

int main()
```

```
{
	// Problem 21.22 - Trapezoidal Rule (Unequally-Spaced)
	vector<double> V = {0.5,2,3,4,6,8,10,11}; // Volume (m^3), V
	vector<double> p = {336,294.4,266.4,260.8,260.5,249.6,193.6,165.6}; // Pressure (kPa), p(V)
	double ps = 0; // Total Pressure (kPa), ps
	int n = V.size()-1;

	std::cout << "Chapter 21 - Problem 21.22" << std::endl;
	std::cout << "============================================" << std::endl;
	std::cout << "Work (kJ), W = ps" << std::endl;
	std::cout << "Total Pressure (kPa), ps" << std::endl;
	std::cout << "Number of Subintervals, n = " << n << std::endl;
	std::cout << "********************************************" << std::endl;

	double W = Trapun(V,p,n);
	std::cout << "Work (kJ), W = " << setprecision(7) << W << std::endl;

	std::cout << "++++++++++++++++++++++++++++++++++++++++++++" << std::endl;
	cout << "\n";

	// Problem 24.4 - Simpson's Rules (Unequally-Spaced)
	vector<double> t = {0,10,20,30,35,40,45,50}; // Time (min), t
	vector<double> c = {10,35,55,52,40,37,32,34}; // Mass Concentration (mg/m^3), c
	const double Q = 4; // Flow Rate Constant (m^3/min), Q
	vector<double> Qc;
	for (int i = 0; i < c.size(); i++)
		Qc.push_back(Q*c[i]);
	double cs = 0; // Total Mass Concentration (mg/m^3), cs
	int m = t.size()-1;

	std::cout << "Chapter 24 - Problem 24.4" << std::endl;
	std::cout << "============================================" << std::endl;
	std::cout << "Mass (mg), M = Q * cs" << std::endl;
	std::cout << "Flow Rate Constant (m^3/min), Q = " << Q << std::endl;
	std::cout << "Total Mass Concentration (mg/m^3), cs" << std::endl;
	std::cout << "Number of Subintervals, n = " << m << std::endl;
	std::cout << "********************************************" << std::endl;

	double M = Uneven(m,t,Qc);
	std::cout << "Mass (mg), M = " << setprecision(7) << M << std::endl;

	std::cout << "++++++++++++++++++++++++++++++++++++++++++++" << std::endl;
	cout << "\n";

	system("pause");
	return 0;
}
```

Problem 7 – Romberg Integration (SCILAB)

```
ans  =


    1.
-->//Example of a script file used to run function Romberg
-->//Before running this program make sure that functions
-->//TrapEq and Romberg are loaded
-->//Problem22_2 --
-->a = 1; b = 2; maxiter = 50; ea = 0.5; //ea in percent
-->// define function to integrate as an inline function
-->function [y]=f222(x)
-->    y = (2*x+3/x)^2;
-->endfunction;
-->//Integrate using Romberg:
-->[I1,n1,iter1,ea1] = Romberg(a,b,maxiter,ea,f222)
 ea1  =


    0.0097823
 iter1  =


    2.
 n1  =


    4.
 I1  =


    25.834565
-->//Problem22_3 --
```

```
-->a = 0; b = 2; maxiter = 50; ea = 0.5; //ea in percent
-->// define function to integrate as an inline function
-->function [z]=f223(t)
-->    z = exp(t)*sin(t)/(1+t^2);
-->endfunction;
-->//Integrate using Romberg:
-->[I2,n2,iter2,ea2] = Romberg(a,b,maxiter,ea,f223)
 ea2  =

    0.0997471
 iter2  =

    2.
 n2  =

    4.
 I2  =

    1.941836
-->//End of script
-->diary(0)
```