```
10
   11
              float z(float x,float y);
   13
              float R(float A, float beta);
   14
              float X(float D, float alpha);
              float v(float R, float g, float theta);
   15
   16
             float e(float a, float k);
              float f(float k);
   18
             float integral (float a, float b, float n);
   19
   20
             void main (void)
   21
   22
          ₽ {
   23
  24
25
                      SYSTEM Initialize();
                       float pi=4.0*atanf(1.0);
                       float originala, originalb, originalc, originald, originale, originalf, originalg;
   26
   27
              float x=2.25,y=3.72,A=4.27,beta=35.0*pi/180,D=6.85,alpha=33.0*pi/180,R1=2.15,g=9.81,theta=43.0*pi/180,a=0.5,k=20,k2=100,a2=0,b=2,n=200;
   28
                      originala= z(x,y);
  29
30
                      originalb= R(A, beta);
                      originalc= X(D,alpha);
   31
                      originald= v(R1,g,theta);
   32
                      originale= e(a,k);
   33
                      originalf= f(k2);
                      originalg= integral(a2,b,n);
   34
   35
                      printf("\n\r\n\r original answers in order are as follows: \n\r %f %f %f %f %f %f %f, originala, originalb, originalc, originald, originalc, or
   36
   37
37
  38
                        float newa, newb, newc, newd, newe, newf, newg;
                        float x1=4.13,y1=2.72, A1=17.25,beta1=15.0*pi/180,D1=1.26,alpha1=19.0*pi/180,R2=15.13,theta1=21.0*pi/180,al=0.75,K3=25,K4=50,
   39
                       newa=z(x1,y1);
   40
                        newb=R(A1,beta1);
   41
                       newc=X(D1,alphal);
   42
   43
                        newd=v(R2,g,theta1);
   44
                       newe=e(a1.k3);
   45
                       newf=f(k4);
   46
                        newg=integral(a3,b1,n2);
   47
                        printf("\n\r The new values calculated are as follows: \n\r %f %f %f %f %f %f, newa, newb, newc, newd, newe, newf, newg);
   48
   49
   50
                                 // Add your application code
   51
   52
   53
   54
              float z(float x, float y)
          ₽ {
   55
   56
                        float 7:
   57
                        z= sqrt(pow(x,2)+pow(y,2));
   58
                        return z;
   59
              float R(float A, float beta)
   61
          ₽ {
   62
                        R= A*cos(beta);
   63
  64
                        return R;
```

```
emain.c x main.c x main.c x
Source History 💼 | 🚱 💀 🔻 🔻 🞝 🖓 🚭 📮 🖟 😓 🔁 🖆 🗐 | 🍏 📵 | 🛍 🚅 🚱
            return z;
 59
       float R(float A, float beta)
    □ {
 61
           float R;
 62
            R= A*cos(beta);
 63
            return R;
  64
 65
 66
 67
        float X(float D, float alpha)
 68 📮 {
            float X;
 69
 70
            X=D*pow(tan(alpha),2);
 71
            return X;
 72
 73
 74
       float v(float R, float q, float theta)
 75 □ {
 76
            float v;
 77
            v= sqrt(R*g*tan(theta));
 78
            return v;
 79
 80
 81
        float e(float a, float k)
 82 🖵 {
 83
            int i=0;
 84
            float sumK=0;
 85
```

```
    © mainc x
    © mainc x

    Source
    History

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □

    □
    □
                                  sumK += pow(0.5,i);
    88
    89
    90
                             return sumK;
    91
    92
    93
                 float f(float k)
    94 🗏 {
    95
                            int 1=0, sumL=0;
    96
    97
                             for(1=0;1<=k;1++)
    98
    99
                                      sumL += 1;
  100
  101
                             return sumL;
  102
  103
  104
                   float integral(float a, float b, float n)
  105 □ {
  106
                            int i=0;
  107
  108
  109
                             float deltax= (b-a)/n; //delta x for the approximated integral solution
  110
                             float X=0, X1=0, Xn=0, sum1=0, sum2=0; // the Simpson's rule terms up to f of Xn
  111
  112
  113
                             for (i=0;i<=n;i++)</pre>
  114
                                      X1=0.5*i;
  115
₩ » @ x »
```

```
∰ main.c × ∰ main.c × ∰ main.c ×
Source History 💼 🔯 👼 + 👼 + 💆 🚭 🚭 📮 😭 🗞 🗞 🖭 💇 🔘 🔠 💇 🚅 🔡
112
113
           for (i=0; i<=n; i++)</pre>
114
115
                X1=0.5*i;
                if(i==0)
116
117
118
                   X1 = 0.5*a;
119
120
                }
121
                else
122
123
                             if(i==n)
124
125
                                 Xn=0.5*b;
126
127
                                  else
128
                                          if(i%2 == 0)
129
130
131
                                              sum1 += 2*0.5*(a+(i*(b-a)/n));
132
                                           }
133
                                          else
134
                                           {
                                              sum2 += 4*0.5*(a+(i*(b-a)/n));
135
136
137
138
139
```

```
Source History 👚 👺 🖫 - 🖫 - 🔍 🔁 🗗 📮 😭 - 🌭 - 🖫 - 朢 🔐 🔐 🔐
127
                                        else
128
129
                                                  if(i%2 == 0)
130
                                                   {
                                                       sum1 += 2*0.5*(a+(i*(b-a)/n));
131
132
                                                  }
133
                                                  else
134
                                                   {
135
                                                       sum2 += 4*0.5*(a+(i*(b-a)/n));
136
137
138
139
140
141
142
              float area, sum3;
143
              sum3 = X1 + Xn + sum1 + sum2;
144
              area = sum3* deltax/3;
145
              return area;
146
147
148
149
150
151 🖵 /*
152 | End of File
₽ COM4 - PuTTY
swers in order are as follows:
4.347517 -3.858764 38844.816406 nan 1.999999 5050.000000 1.333333
The new values calculated are as follows:
original answers in order are as follows:
Part a's answer is: 4.347517
Part b's answer is: 3.497779
Part d's answer is: 4.434878
original answers in order are as follows:
4.945230 16.662221 0.149387 nan 2.000000 1275.000000 2.674048
original answers in order are as follows:
4.945230 16.662221 0.149387 7.548188 2.000000 1275.000000 2.674048
```

main.c x main.c x main.c x

Had originally not mad the angles in radians that's why the terminal has previous solutions with errors. The correct solutions are in the last two lines.