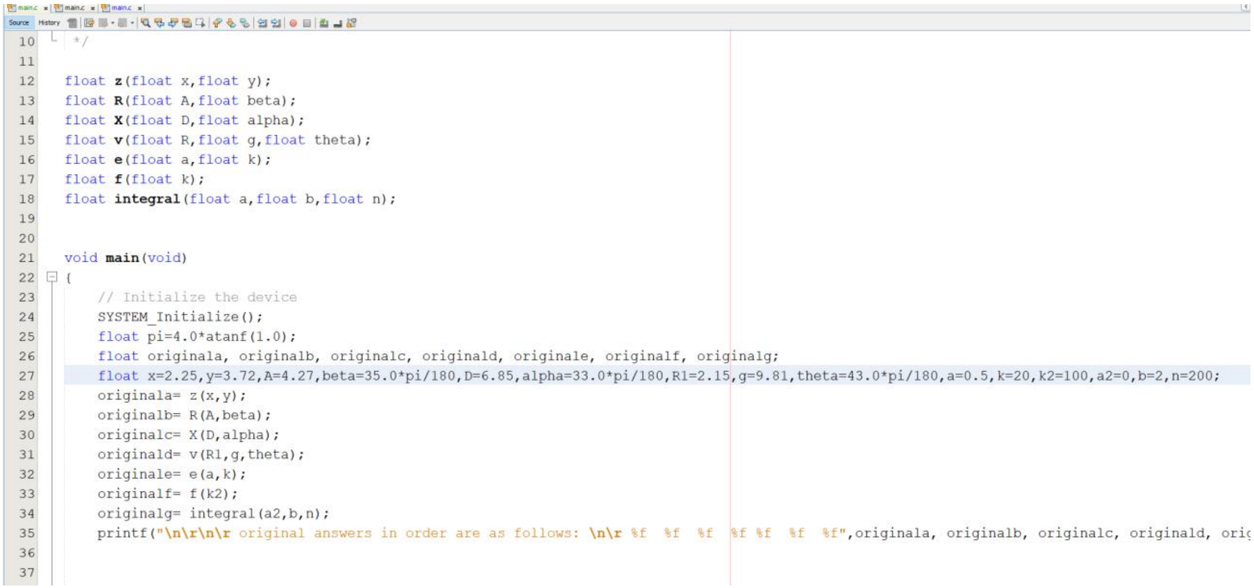
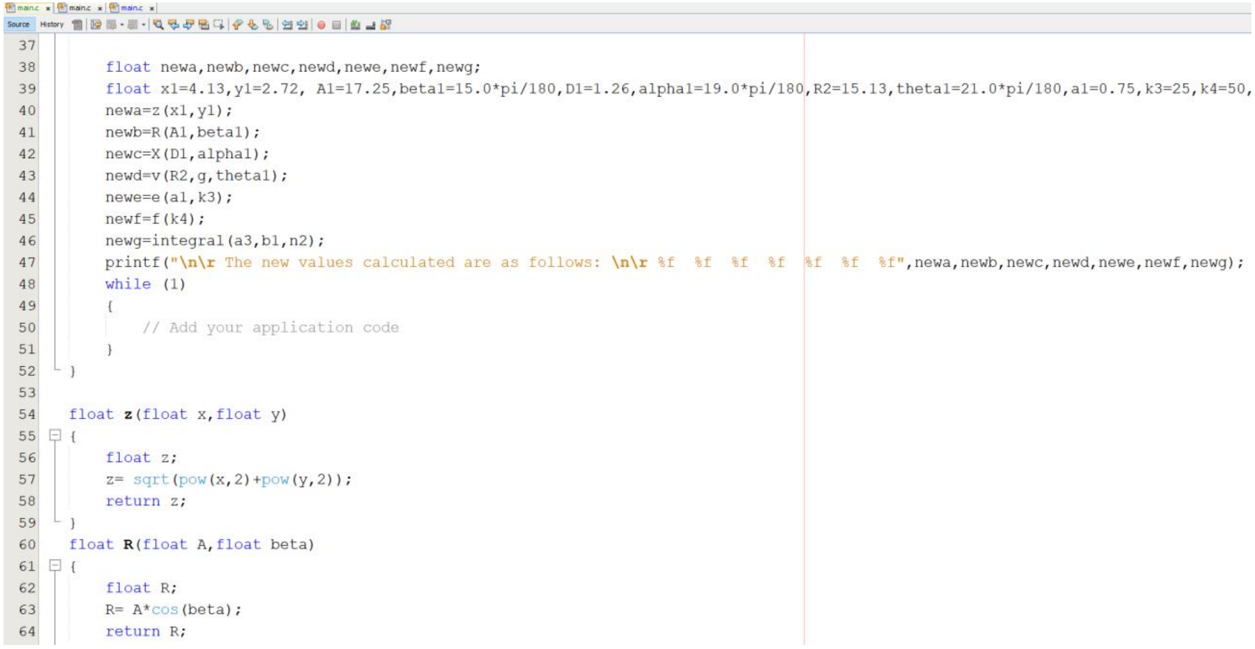
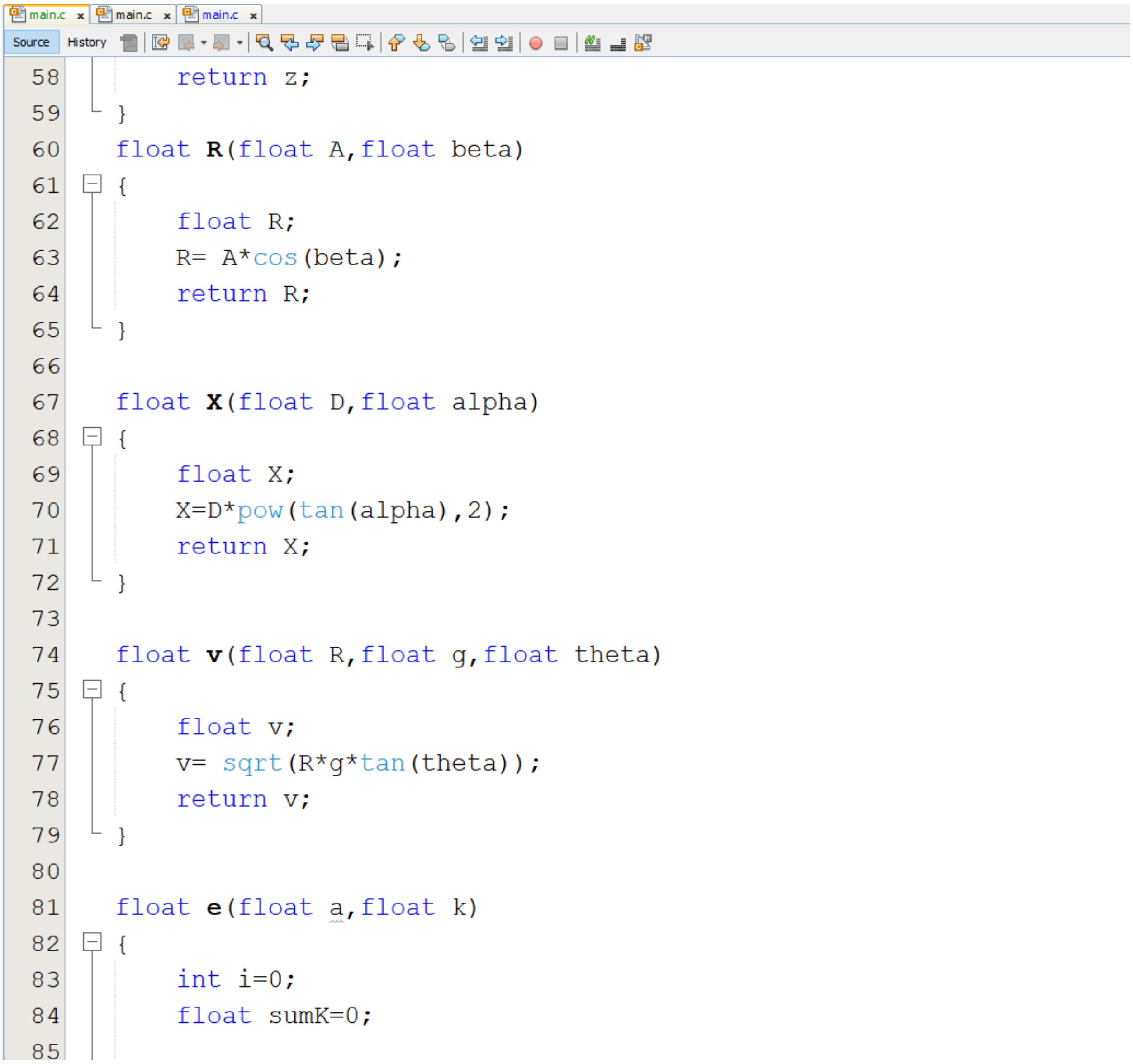


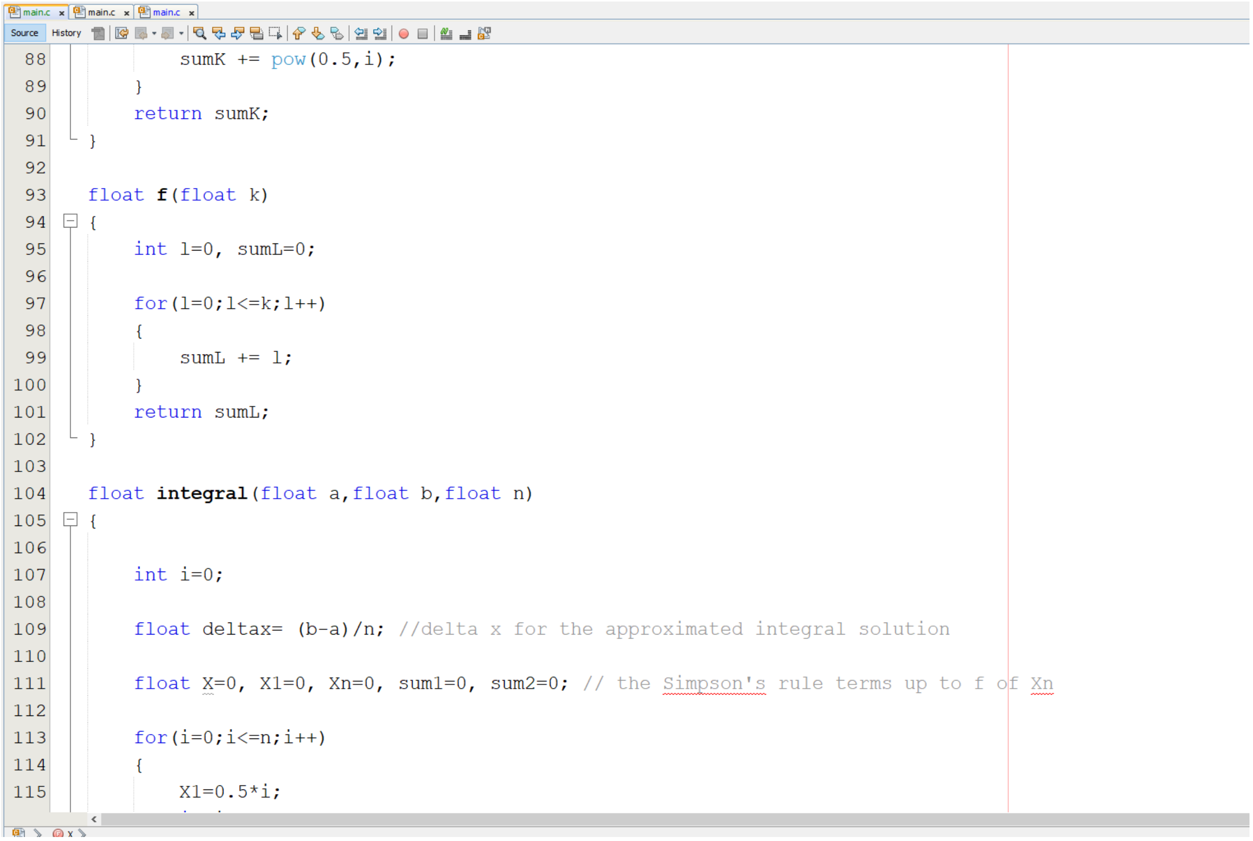
```c
   */


float z(float x,float y);
float R(float A,float beta);
float X(float D,float alpha);
float v(float R,float g,float theta);
float e(float a,float k);
float f(float k);
float integral(float a,float b,float n);


void main(void)
{
    // Initialize the device
    SYSTEM_Initialize();
    float pi=4.0*atanf(1.0);
    float originala, originalb, originalc, originald, originale, originalf, originalg;
    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;
    originala= z(x,y);
    originalb= R(A,beta);
    originalc= X(D,alpha);
    originald= v(R1,g,theta);
    originale= e(a,k);
    originalf= f(k2);
    originalg= integral(a2,b,n);
    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, orig


```



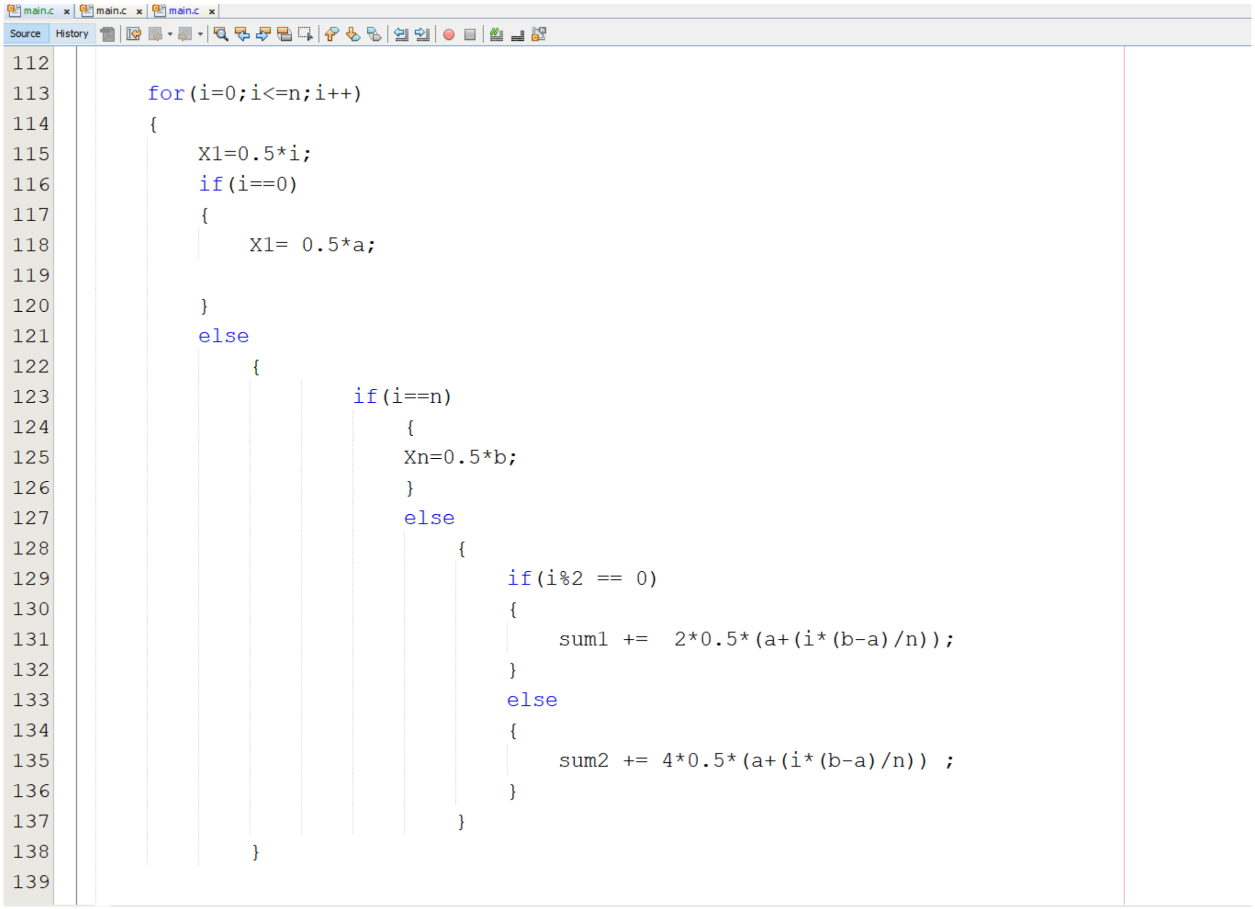
```c

    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,a1=0.75,k3=25,k4=50,
    newa=z(x1,y1);
    newb=R(A1,beta1);
    newc=X(D1,alpha1);
    newd=v(R2,g,theta1);
    newe=e(a1,k3);
    newf=f(k4);
    newg=integral(a3,b1,n2);
    printf("\n\r The new values calculated are as follows: \n\r %f  %f  %f  %f  %f  %f  %f",newa,newb,newc,newd,newe,newf,newg);
    while (1)
    {
        // Add your application code
    }
}

float z(float x,float y)
{
    float z;
    z= sqrt(pow(x,2)+pow(y,2));
    return z;
}
float R(float A,float beta)
{
    float R;
    R= A*cos(beta);
    return R;
```



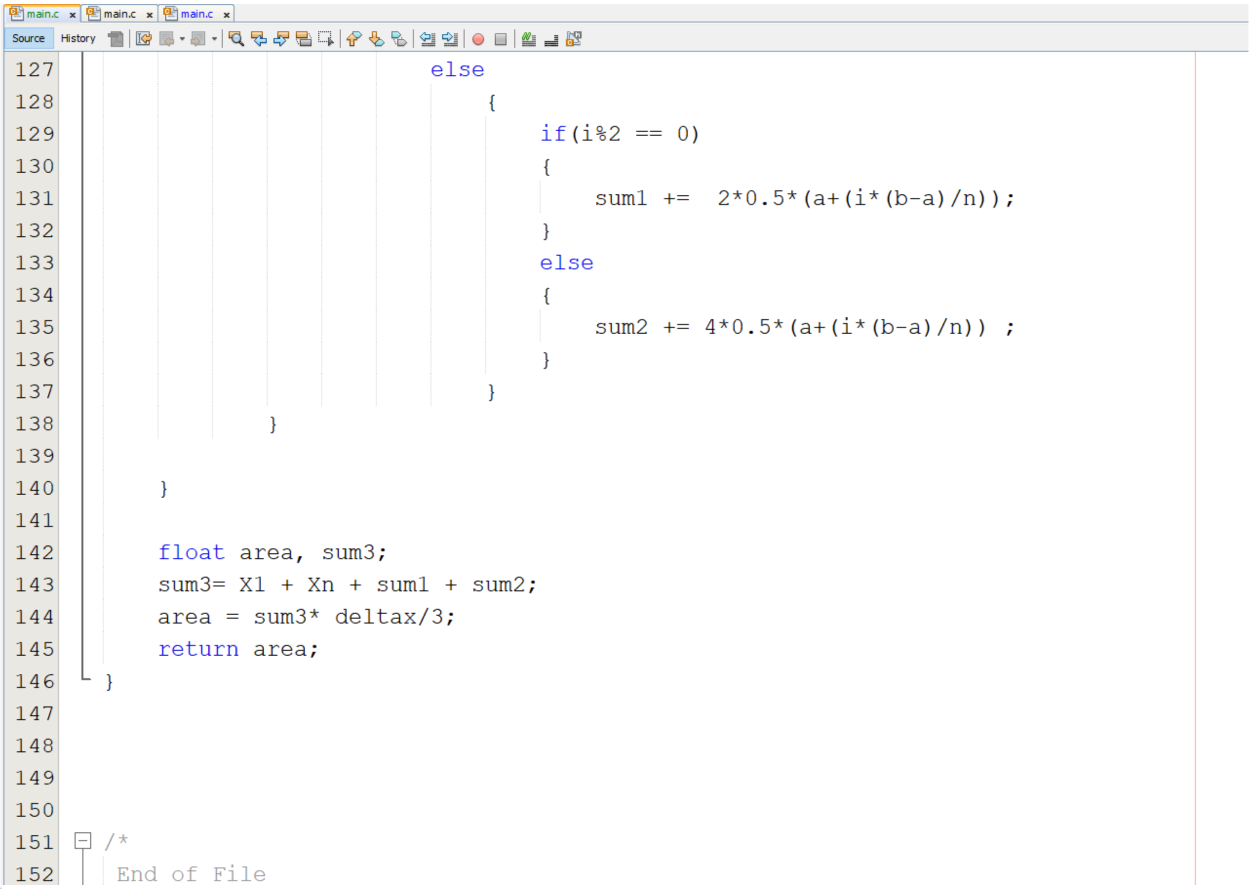
```
    return z;
}
float R(float A,float beta)
{
    float R;
    R= A*cos(beta);
    return R;
}

float X(float D,float alpha)
{
    float X;
    X=D*pow(tan(alpha),2);
    return X;
}

float v(float R,float g,float theta)
{
    float v;
    v= sqrt(R*g*tan(theta));
    return v;
}

float e(float a,float k)
{
    int i=0;
    float sumK=0;

```



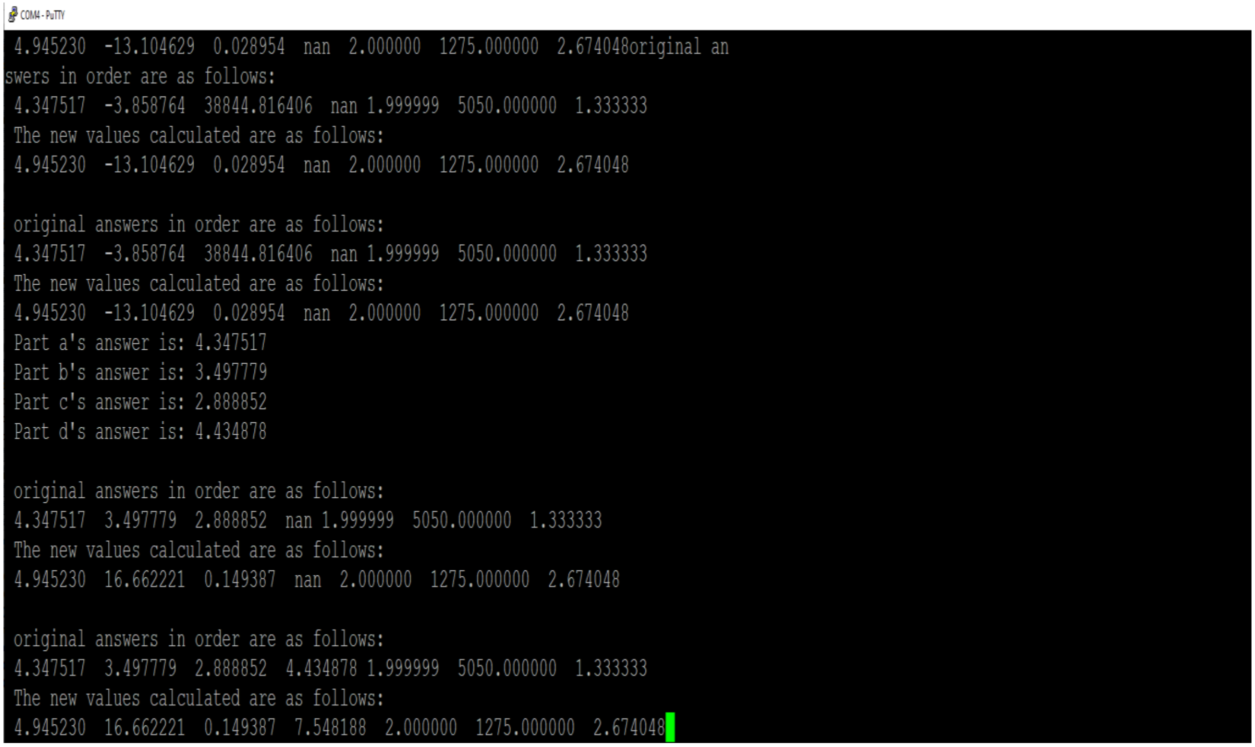
```
        sumK += pow(0.5,i);
    }
    return sumK;
}

float f(float k)
{
    int l=0, sumL=0;

    for(l=0;l<=k;l++)
    {
        sumL += l;
    }
    return sumL;
}

float integral(float a,float b,float n)
{

    int i=0;

    float deltax= (b-a)/n; //delta x for the approximated integral solution

    float X=0, X1=0, Xn=0, sum1=0, sum2=0; // the Simpson's rule terms up to f of Xn

    for(i=0;i<=n;i++)
    {
        X1=0.5*i;
```



```c

    for(i=0;i<=n;i++)
    {
        X1=0.5*i;
        if(i==0)
        {
            X1= 0.5*a;

        }
        else
            {
                    if(i==n)
                        {
                        Xn=0.5*b;
                        }
                        else
                            {
                                if(i%2 == 0)
                                {
                                    sum1 +=  2*0.5*(a+(i*(b-a)/n));
                                }
                                else
                                {
                                    sum2 += 4*0.5*(a+(i*(b-a)/n)) ;
                                }
                            }
            }

```



```
                        else
                            {
                                if(i%2 == 0)
                                {
                                    sum1 +=  2*0.5*(a+(i*(b-a)/n));
                                }
                                else
                                {
                                    sum2 += 4*0.5*(a+(i*(b-a)/n)) ;
                                }
                            }
            }

    }

    float area, sum3;
    sum3= X1 + Xn + sum1 + sum2;
    area = sum3* deltax/3;
    return area;
}




/*
 End of File
```



```
 4.945230  -13.104629  0.028954  nan  2.000000  1275.000000  2.674048original an
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:
 4.945230  -13.104629  0.028954  nan  2.000000  1275.000000  2.674048

 original answers 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:
 4.945230  -13.104629  0.028954  nan  2.000000  1275.000000  2.674048
 Part a's answer is: 4.347517
 Part b's answer is: 3.497779
 Part c's answer is: 2.888852
 Part d's answer is: 4.434878

 original answers in order are as follows:
 4.347517  3.497779  2.888852  nan 1.999999  5050.000000  1.333333
 The new values calculated are as follows:
 4.945230  16.662221  0.149387  nan  2.000000  1275.000000  2.674048

 original answers in order are as follows:
 4.347517  3.497779  2.888852  4.434878 1.999999  5050.000000  1.333333
 The new values calculated are as follows:
 4.945230  16.662221  0.149387  7.548188  2.000000  1275.000000  2.674048
```

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.