→ Find the decimal equivalent of $(1101011)_2$.

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	64	32	16	8	4	2	1
binary number	1	1	0	1	0	1	1

decimal value $\quad 64 + 32 + 8 + 2 + 1 = 107$

$\therefore (1101011)_2 \Longleftrightarrow (107)_{10}$

→ Find the binary equivalent of $(311)_{10}$.

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
	256	128	64	32	16	8	4	2	1	
binary digits	1	0	0	1	1	0	1	1	1	
cummulative sum	256	256	256	288	304	304	308	310	311	(OK)

$\therefore (100110111)_2 \Longleftrightarrow (311)_{10}$

→ Consider a numbering system with 8 digits – a base 8 system. What is the equivalent of $(311)_{10}$ in this system?

	8^3	8^2	8^1	8^0	
	512	64	8	1	
base 8 digits	0	4	6	7	
	→	256	48	7	
cummulative sum	0	256	304	311	(OK)

$\therefore (467)_8 \Longleftrightarrow (311)_{10}$

use of "integral" function | numerical integration

note order

Part b analytical soln

$I_2 = \int_0^2 t e^{-t} dt \Rightarrow$ use integration by part

let $u = t$ $dv = e^{-t} dt$

$du = dt$ $v = -e^{-t}$

where $\int_a^b u\, dv = uv \Big|_a^b - \int_a^b v\, du$

$\therefore \; I_2 = t e^{-t} \Big|_0^2 - \underbrace{\int_0^2 -e^{-t} dt}_{e^{-t} \Big|_0^2}$

$= -(1+t) e^{-t} \Big|_0^2$

$= -3e^{-2} + 1 = \boxed{1 - 3e^{-2}}$ exact ans

Part a analytical soln

$I_1 = \int_0^2 (3x^2 + 1) dx$

$= \left(\frac{3x^3}{3} + x \right) \Big|_0^2 = \left(x^3 + x \right) \Big|_0^2$

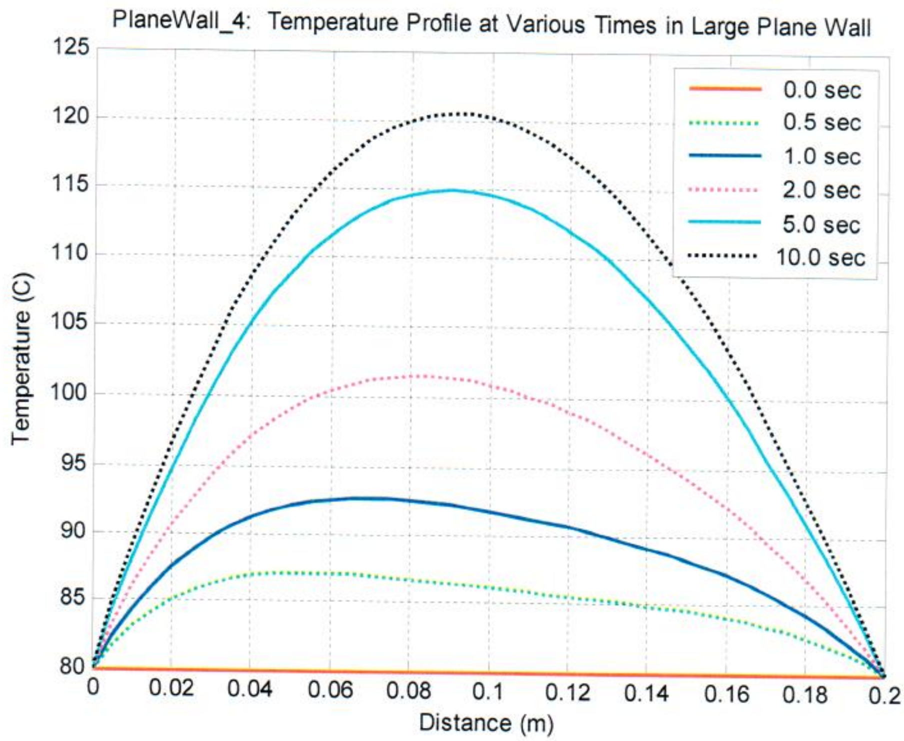
$= 8 + 2 - 0 = \boxed{10}$ exact ans

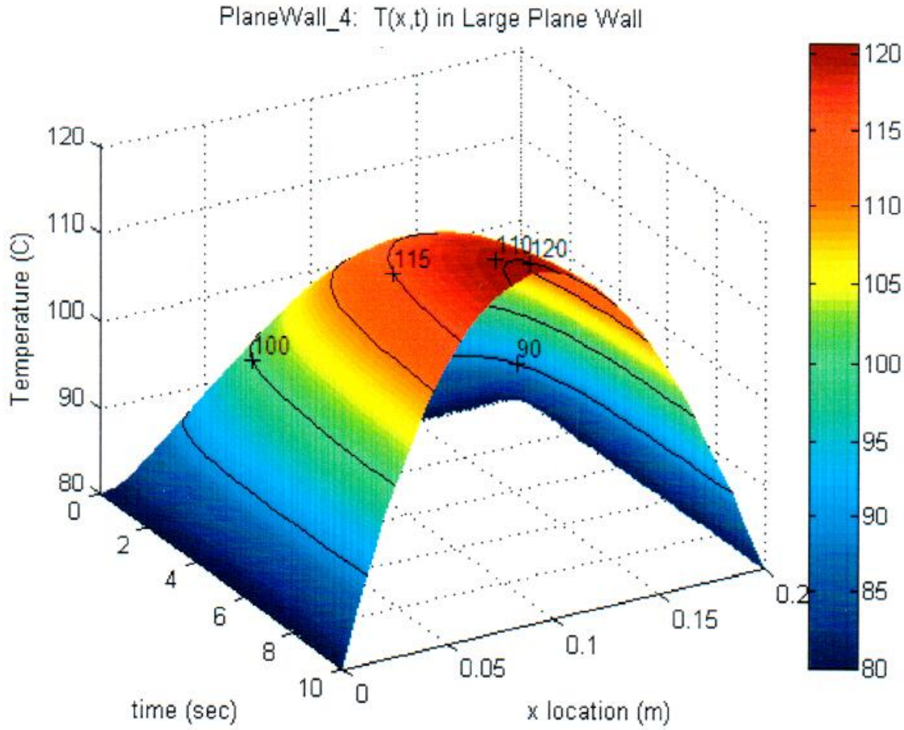
```
%
%   integral_demo.m
%
%   Simply demonstrate the use of the integral function to do numerical integration
%
%   Written by J. R. White, UMass-Lowell  (last update: Oct. 2017)
%

      clear all
%
%   Part a
      fx = @(x) 3*x.*x + 1;      % function of interest
      I1e = 10;                  % analytical soln (via hand calc)
      I1n = integral(fx,0,2);    % numerical soln
      fprintf('\n Part a: The exact value of the integral is %8.6f and the numerical↙
result is %8.6f\n',I1e,I1n);
%
%   Part b
      ft = @(t) t.*exp(-t);      % function of interest
      I2e = 1-3*exp(-2);         % analytical soln (via hand calc)
      I2n = integral(ft,0,2);    % numerical soln
      fprintf('\n Part b: The exact value of the integral is %8.6f and the numerical↙
result is %8.6f\n',I2e,I2n);
%
%   end of program
```

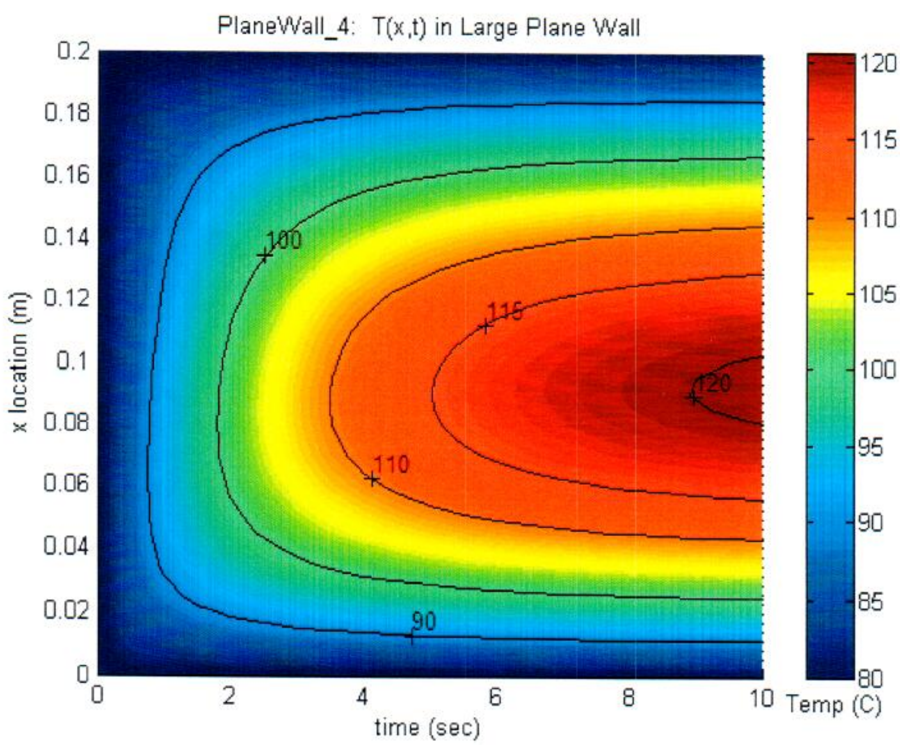
```
>> integral_demo

 Part a: The exact value of the integral is 10.000000 and the numerical result is↙
10.000000

 Part b: The exact value of the integral is 0.593994 and the numerical result is 0.593994
```

Summary Results from PlaneWall_4

```
>> planewall_4
 Needed 6 terms for convergence at t = 0.5 s
 Needed 4 terms for convergence at t = 1 s
 Needed 3 terms for convergence at t = 2 s
 Needed 2 terms for convergence at t = 5 s
 Needed 2 terms for convergence at t = 10 s
```




PlaneWall_4: T(x,t) in Large Plane Wall
Temperature (C)
time (sec)
x location (m)
115
110
120
100
90


PlaneWall_4: T(x,t) in Large Plane Wall
x location (m)
time (sec)
Temp (C)
100
115
120
110
90

```
%
%   PLANEWALL_4.M     Heat Transfer 1-D Planar Wall
%
%   This program implements the analytical solution using the Separation of Variables
%   (SOV) method to a relatively simple heat transfer problem.  The result of the
%   SOV method (see problem description) is an infinite series expansion, and the
%   goal here is to evaluate and plot the T(x,t) distribution given by the infinite
%   series.  In addition, the expansion coefficients in the infinite series are given
%   in integral form, so these will need to be integrated numerically.
%
%   The physical problem models 1-D space-time heat transfer in a large planar wall.
%   The wall has an initial temperature profile that is constant at 80 C and, at time
%   t = 0, an exponentially varying internal heat source is applied with the
%   boundary temperatures remaining constant at 80 C.  The time-independent, but
%   spatially varying heat source simulates a gamma ray flux applied to the left
%   side of the wall (at x = 0).  The object of the Matlab simulation is to compute
%   and visualize the space-time temperature profile within the wall.
%
%   File prepared by J. R. White, UMass-Lowell (last update: Oct. 2017)
%

      clear all,   close all,   nfig = 0;
%
%   problem data
      alf = 0.0015;              % thermal diffusivity  (m^2/s)
      L = 0.2;                   % thickness of plate  (m)
      Qo = 20;                   % heat source at x = 0  (C/s)
      beta = 5;                  % term in exponent of heat source term (1/m)
      TL = 80;                   % left surface temp for all time  (C)
      TR = 80;                   % right surface temp for all time  (C)
      Ti = 80;                   % initial temp throughout wall  (C)
      Nx = 41;                   % number of x values
      x = linspace(0,L,Nx)';     % vector of points to evaluate function
      nmax = 50;                 % max number of nonzero terms in expansion
      tol = 0.001;               % tolerance to stop series evaluation
%
%   anonymous function for steady state solution
      C0 = Qo/(alf*beta*beta);
      C1 = (TR - TL - C0*(1 - exp(-beta*L)))/L;
      C2 = TL + C0;
      wx = @(x) -C0*exp(-beta*x) + C1*x + C2;
%
%   calc eigenvalues and expansion coeffs for n = 1,2,3,...
      ln = zeros(1,nmax);   an =  zeros(1,nmax);
      d = pi/L;
      for n = 1:nmax
        lamN = n*d;    ln(n) = lamN;
        fx = @(x) (Ti-wx(x)).*sin(lamN*x);
        an(n) = (2/L)*integral(fx,0,L);
      end
%
%   choose type of plot to be produced
      ic = menu('Choose type of plot', ...
               'Show T(x) for several different times (2-D plot)  ', ...
               'Show T(x,t) in a variety of surface plots (3-D plots) ');
%
```

```
%    now evaluate series expansion for several different times
       if ic == 1
         tt = [0 0.5 1 2 5 10];          % use this for simple 2-D plot
       else
         tt = linspace(0,10,21);         % use this for 3-D plots (more time points)
       end
       Nt = length(tt);    T = zeros(Nx,Nt);
%
%    since the initial temperature (at t = 0) is known, let's just set it here
       T(:,1) = Ti*ones(size(x));
%
%    now loop over remaining times
       for j = 2:Nt
         t = tt(j);    cc = -alf*t;    mrerr = 1.0;    n = 0;    vx = zeros(size(x));
         while mrerr > tol    &&    n < nmax
           n = n+1;  vxn = an(n)*exp(cc*ln(n)*ln(n))*sin(ln(n)*x);
           vx = vx + vxn;
           i = find(vx);  rerr = vxn(i)./vx(i);   mrerr = max(abs(rerr));
         end
         T(:,j) = vx + wx(x);
         disp([' Needed ',num2str(n),' terms for convergence at t = ',num2str(t),' s'])
       end
%
       switch ic
%
%    *****  Create 2-D plot  *****
       case 1
%
%    plot curves of T(x,t) for various times
       nfig = nfig+1;  figure(nfig)
       Ncm = 6;    scm = ['r-';'g:';'b-';'m:';'c-';'k:'];    st = zeros(Nt,9);
       for j = 1:Nt
         plot(x,T(:,j),scm(j,:),'LineWidth',2); hold on
         st(j,:) = sprintf('%5.1f sec',tt(j));
       end
       title('PlaneWall\_4:  Temperature Profile at Various Times in Large Plane Wall')
       grid,xlabel('Distance (m)'),ylabel('Temperature (C)')
       legend(char(st))
%
%    *****  Create various 3-D plots  *****
       case 2
%
%      nfig = nfig+1;  h = figure(nfig);   set(h,'Renderer','Zbuffer')
       nfig = nfig+1;  figure(nfig);   colormap(jet);
       surf(tt,x,T), shading interp; colorbar, view(60,30), hold on
       axis('tight')
       title('PlaneWall\_4:  T(x,t) in Large Plane Wall')
       ylabel('x location (m)'),   xlabel('time (sec)')
       zlabel('Temperature (C)'),  hold off
%
%      nfig = nfig+1;  h = figure(nfig);   set(h,'Renderer','Zbuffer')
       nfig = nfig+1;  figure(nfig);   colormap(jet);
       surf(tt,x,T), shading interp; colorbar, view(60,30), hold on
       [cc,hh] = contour3(tt,x,T,[90 100 110 115 120]);
       clabel(cc), set(hh,'EdgeColor','k')
       axis('tight')
```

```
    title('PlaneWall\_4:  T(x,t) in Large Plane Wall')
    ylabel('x location (m)'),   xlabel('time (sec)')
    zlabel('Temperature (C)'),  hold off
%
%   nfig = nfig+1;  h = figure(nfig);   set(h,'Renderer','Zbuffer')
    nfig = nfig+1;  figure(nfig);   colormap(jet)
    surf(tt,x,T), shading interp; colorbar, view(2), hold on
    [cc,hh] = contour3(tt,x,T,[90 100 110 115 120]);
    clabel(cc), set(hh,'EdgeColor','k')
    axis('tight')
    title('PlaneWall\_4:  T(x,t) in Large Plane Wall')
    ylabel('x location (m)'),   xlabel('time (sec)')
    zlabel('Temperature (C)'),  gtext('Temp (C)'),  hold off
%
    end
%   end of program
```