

**This Code is written by Shaumik (20BME163D) and Vedansh  
(20BME158D) with the help of chatgpt**

```
clc,clear all
% Define the domain
Lx = 1; % length of the domain in x-direction
Ly = 1; % length of the domain in y-direction
Nx = 50; % number of grid points in x-direction
Ny = 50; % number of grid points in y-direction
dx = Lx/(Nx-1); % grid spacing in x-direction
dy = Ly/(Ny-1); % grid spacing in y-direction

% Define the thermal properties
k = 385; % thermal conductivity
rho = 8900; % density
cp = 0.385; % specific heat capacity
alpha = k/(rho*cp); % thermal diffusivity

% Define the time steps
tmax = 50; % maximum time
dt = 0.001; % time step
Nt = ceil(tmax/dt); % number of time steps
t = linspace(0,tmax,Nt); % time vector

% Define the initial temperature and boundary conditions
T0 = ones(Nx,Ny)*25; % initial temperature distribution
T0(1,:) = 100; % boundary condition at x=0
T0(Nx,:) = 0; % boundary condition at x=Lx
T0(:,1) = 0; % boundary condition at y=0
T0(:,Ny) = 0; % boundary condition at y=Ly

% Solve the unsteady state 2D conduction equation using the finite difference
method
T = T0;
fig = figure;
for n = 1:Nt
    Told = T;
    for i = 2:Nx-1
        for j = 2:Ny-1
            T(i,j) = Told(i,j) + (alpha*dt/(dx*dx))*(Told(i+1,j)-
2*Told(i,j)+Told(i-1,j)) + (alpha*dt/(dy*dy))*(Told(i,j+1)-2*Told(i,j)+Told(i,j-
1));
        end
    end
    if n < Nt/2
        T(1,:) = 100; % boundary condition at x=0
        T(Nx,:) = 0; % boundary condition at x=Lx
    else
        T(1,:) = 0; % insulated boundary condition at x=0
        T(Nx,:) = 0; % insulated boundary condition at x=Lx
    end
    % Plot the temperature profile as a contour plot
    contourf(T',28);
    colorbar;
    xlabel('x');
    ylabel('y');
    title(['Temperature Profile at t=' num2str(t(n))]);
end
```

```

% Update the figure
drawnow;
% Pause for a short time to show the animation
pause(0.01);
end

% Find the heating and cooling times
heating_time = t(find(T(1,:) == max(T(1,:)),1)) % time to reach maximum
temperature at x=0
cooling_time = t(find(T(Nx,:) == min(T(Nx,:)),1)) % time to cool down to minimum
temperature at x=Lx

```

