```
%Kyle Davis
%December 4 2016

%Simple Geometry

%function HT_Simple
%Clearing all Variable and Closing all Windows
clc
clear
close all
%promping user to attain mesh grid size
prompt = 'What size Grid? Only multiples of 10. ';

%Assigning Prompt Value to n
n= input(prompt);

%Populating Matrices
A = zeros(n^2);
b = zeros(n^2,1);
T = zeros(n^2,1);
type = zeros(n);
%Declaring variables
%Heat Transfer Coeffient [W/(m^2*k)]
h = 34;
%Thermal Conductivity [W/(m*k)]
k = 8;
%Temperature of the fluid surrounding the mesh
Tflu = 8+273;

%Heat generation for nodes within the mesh
qgen = 45000;

%Grid size in meters
x= 1;
y=x;
dx=1/n;
%Boundary condition on top of the Mesh
for i=1:n^2;
Ttop(i) = (12*(dx/2+dx*(i-1))+9)+273;
end
i=n;
%Declaring Node Type
for i =1:n^2
if i ==1
type(1,1)=1;                %Top Left
elseif i ==n
type(1,n)=2;                %Top Right
elseif i==n^2-(n-1)
type(n,1)=3;                %Bottom Left
elseif i>1 && i<n
type(1,2:n-1) =5;           %Top row
elseif i>(n^2-(n-1)) && i < n^2
type(n,2:n-1)=6;            %Bottom Row
elseif mod(i,n)==1 && i~=1 && i~=n*(n-1)+1
type(2:n-1,1)=7;            %Left Edge
elseif mod(i,n)==0 && i~=n && i~=n^2
```

```
type(2:n-1,n)=8;            %Right Edge
elseif i >=5.6*n && i<=7.9*n && mod(mod(i,n),6)==0
type(mod(5.6*n,n):mod(5.8*n,n),mod(7.6*n,n):mod(7.9*n,n))=9;
else
type(n,n)=4;                %Bottom Right

end

end

%Reshaping type matrix to get node values
node = reshape(type',[1,n^2]);

% Filling in Matrix A and B
for i=1:n^2
if node(i)==1
A(i,i) = -(h*dx/k+4);
A(i,i+1) = 1;
A(i,i+n) = 1;
b(i) = -(h*dx/k*Tflu+2*Ttop(i));
elseif node(i) ==2
A(i,i) = -4;
A(i,i-1) = 1;
A(i,i+n) = 1;
b(i)= -2*Ttop(i);
elseif node(i) == 3
A(i,i)=-(h*dx/k+2);
A(i,i+1) =1;
A(i,i-n) =1;
b(i) = -(h*dx/k)*Tflu;
elseif node(i)==4
A(i,i)=-2;
A(i,i-1) =1;
A(i,i-n) = 1;
elseif node(i) == 5
A(i,i)=-5;
A(i,i-1)=1;
A(i,i+1)=1;
A(i,i+n)=1;
b(i)=-2*Ttop(i);
elseif node(i) == 6
A(i,i)=-3;
A(i,i+1)=1;
A(i,i-n)=1;
A(i,i-1)=1;
elseif node(i) ==7
A(i,i)=-((h*dx/k+3));
A(i,i+1)=1;
A(i,i-n)=1;
A(i,i+n)=1;
b(i)=-h*dx/k*Tflu;
elseif node(i) == 8
A(i,i)=-3;
A(i,i-1)=1;
A(i,i-n)=1;
A(i,i+n)=1;
```

```
elseif node(i) == 9
A(i,i)=-4;
A(i,i-1)=1;
A(i,i+1)=1;
A(i,i-n)=1;
A(i,i+n)=1;
b(i)=-qgen*dx^2/k;
else
A(i,i)=-4;
A(i,i-1)=1;
A(i,i+1)=1;
A(i,i-n)=1;
A(i,i+n)=1;
end
end

%Filling the Temperature Matrix
T = A\b;
Temp = reshape ( T,[n,n]);
Temp =Temp.';

%Thermal View
figure(1)
surf(Temp')
view(90,90)
title('Thermal Profile')
xlabel('Distance in X Direction (m)')
ylabel('Distance in y Direction (m)')
x1=0;
figure(2)
surf(Temp')
title('Thermal Profile')
xlabel('Distance in X Direction (m)')
ylabel('Distance in y Direction (m)')
zlabel('Distance in z Direction (m)')

%Plotting Side Profiles
TopT=Temp(1,1:n);
BotT=Temp(n,1:n);
LeftT=Temp(1:n,1);
RightT=Temp(1:n,n);


figure(3)
x1=(dx/2:dx:1-dx/2);
%Bottom Edge
subplot(3,2,1)
plot(x1,BotT)
title('Bottom Edge')
xlabel('Distance (m)')
ylabel('Temperature (K)')

%Left Edge
subplot(3,2,2)
plot(x1,LeftT)
title('Left Edge')
```

```
xlabel('Distance (m)')
ylabel('Temperature (K)')

%Top Edge
subplot(3,2,3)
plot(x1,TopT)
title('Top Edge')
xlabel('Distance (m)')
ylabel('Temperature (K)')

%Right Edge
subplot(3,2,4)
plot(x1,RightT)
title('Right Edge')
xlabel('Distance (m)')
ylabel('Temperature (K)')

Qone=zeros(n,n);
Qtwo=zeros(n,n);
Qthree = zeros(n,n);
Qfour = zeros(n,n);
Qfive = zeros(n,n);
Qsix = Qone;
Qsvn = Qone;
Qate = Qone;
Qnine= Qone;
Qzero = Qone;

%Node Verification
for i=1:n^2
if node(i) ==1
qoneL(i) = (h*dx)*(Tflu-T(i));
qoneR(i) = k*(T(i+1)-T(i));
qoneU(i) = 2*k*(Ttop(i)-T(i));
qoneD(i) = k*(T(i+n)-T(i));
qone(i) = qoneL(i)+qoneR(i)+qoneU(i)+qoneD(i);
Qone(i)= sum(qone);
elseif node(i) == 2
qtwoL(i) = k*(T(i-1)-T(i));
qtwoU(i) = 2*k*(Ttop(i)-T(i));
qtwoD(i) = k*(T(i+n)-T(i));
qtwo(i) = qtwoL(i)+qtwoU(i)+qtwoD(i);
Qtwo(i) = sum(qtwo);
elseif node (i) == 3
qthrL(i) = h*dx * (Tflu-T(i));
qthrR(i) = k*(T(i+1)-T(i));
qthrU(i) = k*(T(i-n)-T(i));
qthr(i) = qthrL(i)+qthrR(i)+qthrU(i);
Qthree(i) = sum(qthr);
elseif node(i) == 4
qfourA(i) = k*(T(i-1)-T(i));
qfourB(i) = k*(T(i-n)-T(i));
qfour(i) = qfourA(i)+qfourB(i);
Qfour(i) = sum(qfour);
elseif node(i)== 5
qfiveL(i) = k*(T(i-1)-T(i));
```

```
qfiveR(i) = k*(T(i+1)-T(i));
qfiveU(i) = 2*k*((Ttop(i))-T(i));
qfiveD(i) = k*(T(i+n)-T(i));
qfive(i) = qfiveL(i)+qfiveR(i)+qfiveU(i)+qfiveD(i);
Qfive(i) = sum(qfive);
elseif node(i) == 6
qsixL(i) = k*(T(i-1)-T(i));
qsixR(i) = k*(T(i+1)-T(i));
qsixU(i) = k*(T(i-n)-T(i));
qsix(i) = qsixL(i)+qsixR(i)+qsixU(i);
Qsix(i) = sum(qsix);
elseif node(i) == 7
qsvnL(i) = h*dx*(Tflu-T(i));
qsvnR(i) = k*(T(i+1)-T(i));
qsvnU(i) = k*(T(i-n)-T(i));
qsvnD(i) = k*(T(i+n)-T(i));
qsvn(i) = qsvnL(i)+qsvnR(i)+qsvnU(i)+qsvnD(i);
Qsvn(i) = sum(qsvn);
elseif node(i) ==8
qateA(i) = k*(T(i-1)-T(i));
qateB(i) = k*(T(i-n)-T(i));
qateC(i) = k*(T(i+n)-T(i));
qate(i) = qateA(i)+qateB(i)+qateC(i);
Qate(i) = sum(qate);
elseif node(i) == 9
qninA(i) = k*(T(i-1)-T(i));
qninB(i) = k*(T(i+1)-T(i));
qninC(i) = k*(T(i-n)-T(i));
qninD(i) = k*(T(i+n)-T(i));
qninE(i) = qgen*(dx)^2;
qnin(i) = qninA(i)+qninB(i)+qninC(i)+qninD(i)+qninE(i);
Qnine(i) = sum(qnin);
elseif node(i) == 0
qzeroA(i) = k*(T(i-1)-T(i));
qzeroB(i) = k*(T(i+1)-T(i));
qzeroC(i) = k*(T(i-n)-T(i));
qzeroD(i) = k*(T(i+n)-T(i));
qzero(i) = qzeroA(i)+qzeroB(i)+qzeroC(i)+qzeroD(i);
Qzero(i) = sum(qzero);

end
end
Qtotal = zeros(n,n);
Qtotal = Qone+Qtwo+Qthree+Qfour+Qfive+Qsix+Qsvn+Qate+Qnine+Qzero;
QT= Qtotal.';
QTverif = abs(reshape(QT,[n^2,1]));

%Quantity of Heat Transfer Verification

for i=1:n^2
if QTverif(i) ~= 0 && QTverif(i) >10^-10
fprintf ('Q in one of the nodes is Greater than 10^10 and not zero');
else
fprintf ('The Program Works!');
end
end
```

```
%Table of Quantity of Heat Transfer per Unit depth
Topdq = QT(1,1:n);
Topdq = sum(Topdq);
Leftdq = QT(1:n,1);
Leftdq = sum(Leftdq);

Names ={'Top Edge';'Left Edge';};
DQ=[Topdq;Leftdq];
T= table(Names,DQ);
disp(T)
```