The governing equation is $\frac{\partial T}{\partial t} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$ and for steady state, we have $\frac{\partial T}{\partial t} = 0$

In[ ]:=
```
nx = 21; ny = 21;
Δx = 1/(nx - 1); Δy = 1/(ny - 1);
k = 1; h = 10; T0 = 300; Tleft = 350;
T = Array["T", {nx, ny}];
```

In[ ]:=
```
discreteEqns = Table[
   (T[[i + 1, j]] - 2 T[[i, j]] + T[[i - 1, j]])/Δx^2 + (T[[i, j + 1]] - 2 T[[i, j]] + T[[i, j - 1]])/Δy^2 == 0,
   {i, 2, nx - 1}, {j, 2, ny - 1}];
```

In[ ]:=
```
leftBoundary = Table[T[[1, j]] == Tleft, {j, 2, ny - 1}];
topBoundary = Table[T[[i, ny]] == T[[i, ny - 1]], {i, 1, nx}];
bottomBoundary = Table[T[[i, 1]] == T[[i, 2]], {i, 1, nx}];
rightBoundary =
  Table[(-k/Δx) (T[[nx, j]] - T[[nx - 1, j]]) == h (T[[nx, j]] - T0), {j, 2, ny - 1}];
```

In[ ]:=
```
eqns = Join[Flatten[discreteEqns],
   leftBoundary, topBoundary, rightBoundary, bottomBoundary];
```

In[ ]:=
```
sol = NSolve[eqns, Flatten[T]];
```

In[ ]:=
```
TVals = T /. sol // #[[1]] &;
```

In[ ]:=
```
ListContourPlot[Transpose[TVals], PlotLegends → Automatic]
```

Out[ ]=