Write a program using pthreads to estimate the temperature of all points on a grid in a similar fashion to the algorithm provided in the parallel computing program: https://computing.llnl.gov/tutorials/parallel_comp/#ExamplesHeat.

You can start out by modifying the sample “sum.c” code, or write your own code from scratch.

You should use the following parameters:

a) Grid size = 1000 x 1000, spanning all points in the square between coordinates (1,1) and (1000,1000).

b) Initial condition: All center points in the region (200, 200) to (800, 800) have a temperature of 500 degrees, and all other points have a temperature of zero. Points outside the grid (i.e., neighbors of points on the boundary) always have a temperature of zero that does not change.

c) At each time step $t$, the temperature of a point at coordinates $(x,y)$ is computed from the temperatures of the neighboring points in the previous time step $(t-1)$ according to the following equation:

$$T(x,y)(t) = T(x,y)(t-1) + C_x \cdot (T(x+1,y)(t-1) + T(x-1,y)(t-1) - 2 \cdot T(x,y)(t-1))$$
$$+ C_y \cdot (T(x,y+1)(t-1) + T(x,y-1)(t-1) - 2 \cdot T(x,y)(t-1))$$

Where $C_x=0.002$ and $C_y=0.005$

d) Run the program for 3000 steps. Note that depending on how you split your data, you may need to communicate information to neighboring processors after each time step.

e) After each 200 time steps, you should print the temperatures of the following points: (1, 1), (200,200), (400, 400), (500, 500), (700, 700), and (900,900).

You should turn in a text copy of your program, a table containing the parallel speedups, and an email attachment with your C program. You will get a better score if you do a good job parallelizing your algorithm to get a high parallel speedup.