Lab 6

What our program does

- The poorly drawn figure on the left shows a cross section of a metal bar (gray), which is covered by a heated wrapper (red) and each end is in very cold liquid (blue)

What our program does

	100	100	100	100	100	100	
-40	0	0	0	0	0	0	-40
-40	0	0	0	0	0	0	-40
-40	0	0	0	0	0	0	-40
-40	0	0	0	0	0	0	-40
-40	0	0	0	0	0	0	-40
-40	0	0	0	0	0	0	-40
	100	100	100	100	100	100	

- We can simulate the effect this has on the temperature of the bar by representing the bar as an array, with the top an bottom elements initialized to 100 to represent the wrapper, and the ends initialized to -40, to represent the cold liquid.

- The white elements at the corner are not accessed, and can be any value.

We then iterate over the array repeatedly, updating the gray elements with the loop below until the value of a converges.

```
for (int 1=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        a[i][j] = (a[i][j] + a[i][j] + a[i][j] + a[i][j] + a[i][j])/5.0;
    }
}
```

Making it parallel

	100	100	100	100	100	100		**Row 0**
-40	0	0	0	0	0	0	-40	**Row 1**
-40	0	0	0	0	0	0	-40	**Row 2**
-40	0	0	0	0	0	0	-40	**Halo 1**

P0

-40	0	0	0	0	0	0	-40	**Halo 2**
-40	0	0	0	0	0	0	-40	**Row 3**
-40	0	0	0	0	0	0	-40	**Row 4**
-40	0	0	0	0	0	0	-40	**Halo 3**

P1

-40	0	0	0	0	0	0	-40	**Halo 4**
-40	0	0	0	0	0	0	-40	**Row 5**
-40	0	0	0	0	0	0	-40	**Row 6**
	100	100	100	100	100	100		**Row 7**

P2

- When we make it parallel, we will split the array P sections, where P is the number of processors. Shown at the left are the array divided into 3 sections.

- The green squares are *halo* regions. These are extra array rows added for data to be received from other processors.

- Halo 1 on P0 will get data from Row 3 on P1. Halo 2 on P1 will get data from Row 2 on P0. Halo 3 on P1 will get data from Row 5 on P2.

- **You will need to add communication code to make this happen. The places you will add the communication code are indicated by comments.**

We then iterate over the array repeatedly, updating the gray elements with the loop below until the value of a converges.

```
for (int 1=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        a[i][j] = (a[i][j] + a[i][j] + a[i][j] + a[i][j] + a[i][j])/5.0;
    }
}
```