Name: ____________________________

Directions: **Work only on this sheet** (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing.

Group, in-class programming problem:

You are given an image in which each pixel is known to be either fully black (1) or fully white (0), and in which the black sections are in the form of nonoverlapping squares. The task is to find all of the squares. You must use either Rdsm or OpenMP.

The form of function call, for R/Rdsm, is

```
findsqrs <- function(m)
```

where **m** is the matrix of pixels. The return value is a three-column matrix, where a row **(u,v,w)** means that the "northwest corner" of a square is at row **u**, column **v**, and the width is **w**.

The C/OpenMP form is

```
int *findsqrs(int *m, int mr, int mc)
```

Here **mr** and **mc** are the number of rows and columns in **m**, and the return value is three-column matrix as above.

Row and column numbering will increase as one goes down and rightward in the image, starting at (1,1) for R and (0,0) for C. A square could be of size 1x1.

As an example (for R), consider this matrix:

```
> m
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]    0    0    0    0    0    0    0    0    0
[2,]    0    0    1    1    1    0    0    0    0
[3,]    0    0    1    1    1    0    0    0    0
[4,]    0    0    1    1    1    0    1    1    0
[5,]    0    1    1    0    0    0    1    1    0
[6,]    0    1    1    0    0    0    0    0    0
```

The function **findsqrs()** would return the matrix

```
     [,1] [,2] [,3]
[1,]    2    3    3
[2,]    4    7    2
[3,]    5    2    2
```

(possibly with the rows permuted).

Requirements:

- A point (u,v) will not be tested for "NWCness" (being the northwest corner of a square) by more than one thread.
- Once a square has been discovered, no thread will test any points within it for NWCness.

At the end of the class period, e-mail me your code (all routines in a single file). Of course, make sure all the names of your group members are in comments at the top of the file.

If your code doesn't run properly, just send me what you have, with some comments added as to what you would do to try to fix it if you had more time.

I am doubtful that the most straightfoward implementation will actually produce a speedup, at least in R. However, your first priority must be to **write code that works**. If you get it running and would like to submit a second version that seems faster, please do so and you will receive Extra Credit.