Due Date: Homework is due on Monday Feb 26 before class.

You will have to finish writing the subroutines in order to do make the code actually work. Remember that you are not allowed to change the form of the subroutines and you should only modify the code where it says "COMPLETE HERE"!

I. LATTICE CLASS

The program takes in as input: Nlin, pr, Nclust, Nbin, SEED and latt_. The variable Nlin is the linear size of the 2-D lattice, pr is the site activation probability and SEED is the seed for the random number generator. We do not need pr, SEED, Nclust and Nbin for the homework this week. Assign them values "0" in the param.dat file for now.

A. LATTICE::LATTICE

This is the constructor for the LATTICE class. It is here that all the properties of the lattice defined in the declaration of the LATTICE class should be defined. **nrnbrs** is a member of the LATTICE class. This table stores the nearest neighbors of each site on the square lattice. Depending on whether the program is called with "sqlatt_OBC" or "sqlatt_PBC" for the latt_ parameter in "param.dat", the program should create the appropriate table for an arbitrary input value of $N_{\rm lin}$. This information is stored in an array of arrays. The first element of the array runs over the Nsite lattice sites **and the second element over the number of nearest neighbors** of the lattice site associated with the first element (for the PBC case there are always four neighbors, for the OBC case it depends on the lattice site). Fill up the neighbors using the following ordering convention: $+x, +y, -x, -y$ (if the neighbor is absent just skip those elements)
and then the value stored in the matrix entry is the neighbor number (i.e. $nrnbrs[i][j]$ is the j^{th} neighbor of the i^{th} site). If that was too terse think see Fig. 2 for output of the latt.print() function.

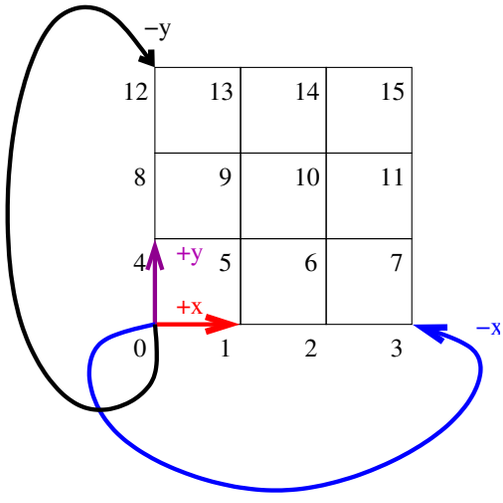

FIG. 1: Our convention for the site numbering on an $N_{\rm lin} = 4$, i.e. 4×4 lattice. Also shows as an example how the 4 nearest neighbors of the site "0" are labeled for PBC.

You can debug this part of the program by calling the member of the LATTICE class called print(). This function has been given to you already.

Finish writing the LATTICE::LATTICE function and debug it. Include a convincing explanation, complete with a print out of the data of your program, of how you decided the code was debugged *(40 points)*.

```
--- Parameters at input for percolation problem ---
Nlin = 4; prob. of site = 0.1
Number of clusters in a bin = 0; Number of bins = 0
RNG will be given SEED of = 10
Percolation problem on lattice --> sqlatt_OBC
---printing out properties of lattice ---
size is  4x4
neighbors are
0 : [nn of nbrs = 2] which are  1 4
1 : [nn of nbrs = 3] which are  2 5 0
2 : [nn of nbrs = 3] which are  3 6 1
3 : [nn of nbrs = 2] which are  7 2
4 : [nn of nbrs = 3] which are  5 8 0
5 : [nn of nbrs = 4] which are  6 9 4 1
6 : [nn of nbrs = 4] which are  7 10 5 2
7 : [nn of nbrs = 3] which are  11 6 3
8 : [nn of nbrs = 3] which are  9 12 4
9 : [nn of nbrs = 4] which are  10 13 8 5
10 : [nn of nbrs = 4] which are  11 14 9 6
11 : [nn of nbrs = 3] which are  15 10 7
12 : [nn of nbrs = 2] which are  13 8
13 : [nn of nbrs = 3] which are  14 12 9
14 : [nn of nbrs = 3] which are  15 13 10
15 : [nn of nbrs = 2] which are  14 11
```

(a)OBC

```
--- Parameters at input for percolation problem ---
Nlin = 4; prob. of site = 0.1
Number of clusters in a bin = 0; Number of bins = 0
RNG will be given SEED of = 10
Percolation problem on lattice --> sqlatt_PBC
---printing out properties of lattice ---
size is  4x4
neighbors are
0 : [nn of nbrs = 4] which are  1 4 3 12
1 : [nn of nbrs = 4] which are  2 5 0 13
2 : [nn of nbrs = 4] which are  3 6 1 14
3 : [nn of nbrs = 4] which are  0 7 2 15
4 : [nn of nbrs = 4] which are  5 8 7 0
5 : [nn of nbrs = 4] which are  6 9 4 1
6 : [nn of nbrs = 4] which are  7 10 5 2
7 : [nn of nbrs = 4] which are  4 11 6 3
8 : [nn of nbrs = 4] which are  9 12 11 4
9 : [nn of nbrs = 4] which are  10 13 8 5
10 : [nn of nbrs = 4] which are  11 14 9 6
11 : [nn of nbrs = 4] which are  8 15 10 7
12 : [nn of nbrs = 4] which are  13 0 15 8
13 : [nn of nbrs = 4] which are  14 1 12 9
14 : [nn of nbrs = 4] which are  15 2 13 10
15 : [nn of nbrs = 4] which are  12 3 14 11
```

(b)PBC

FIG. 2: Output of "latt.print()" function. Neighbors for $N_{\text{lin}} = 4$, i.e. 4×4 lattice