Maddy Allan-Rahill
Bella Virgilio
Leslie Fuentes
Final Project: Four in a Row

2. A **Technical Report** with description of:

2a. List the **ADTs** that will be used and what information they will store. Include a brief justification for each of your choices.

The two ADTs that we will use are a graph and a linked list. We will be using linked lists to implement the graph structure. We chose to use a graph to represent the game board, which will contain the discs, because we can easily test for connectivity between discs to determine whether there are 4 discs in a row. The linked lists will store these connections.

2b. A list of the important **classes** that you expect to define for your project with a brief description of the purpose of each class. Some of these classes should capture the basic objects that exist in the problem. There may also be classes that embody the graphical user interface, or the main() method. This list should include the classes that implement the ADTs that you plan to use. Note that as you proceed with your program development, you may discover other classes that would be useful to define for your application.

Disc class - Creates a game disc that will be used to play within the board
Player class - Creates a player that will utilize a disk to make moves on the board
Board class - Creates a board with specific parameters that will hold discs set by the players; implements Graph and LinkedList ADT.
Game class - Represents a game of Connect Four
GUI class - Creates a GUI to display the board and discs set by the players
Driver class - test code and create instances of Connect 4 game
Panel class - will manage the GUI to represent what is happening within the Game class

2c. A list of some of the main **actions** that you expect to be embodied in methods in your new class definitions (you do not need to include the basic operations defined for the ADT classes that you plan to use). As you proceed with your program development, you will probably discover additional useful methods to define for various classes.

Disc class
getPosition() - gets the position of the disc

setPosition(column, row) - sets the position of the disc at the specified parameters
getColor() - gets the color of the disc
setColor(color) - sets the color of the disc
toString() - string representation of a disc

Board class
clearBoard() - empties the board
addDisc(column) - drops a disc into the board into a specified column
isConnected(disc) - returns true if two discs are connected
toString() - string representation of the board

Player class
setName(name) - sets a name to the current player
getName() - returns the name of the current player
getColor() - returns the color designated to the player
setColor(color) - sets a color to specific player
toString()

Game class
playConnectFour() - begins a game of Connect Four
getTurn() - displays whose turn it is
bestOfThree() - will store the scores of the players Best of Three game
toString()
win() - determines which player successfully placed 4 disks in a row

Driver class
main() - where we will make instances of Game and test our code

GUI class
main() - create a GUI to display board game