Prompt A:

The single responsibility principle is the the idea that every block of code should have only one responsibility. That way your program is completely modular. If you ever want to perform a single action you will always perform only that action and nothing more. This also makes it easier to re-use code in future programs.

An example of the single responsibility principle would be a method that asks the user for input. That could then return a value into a method that processes input and if that input matches certain values it would call certain methods. Each method that could be called would first call the method that gets input then process that input in some form.

Prompt B:

Parse attempts to read a piece of data and convert it to another type. If it succeeds in converting the data it will return the now converted data. If it fails to convert it will throw an exception. TryParse does the same as a normal parse but instead of returning the data, it outs the data if the conversion was successful. It then returns a bool value, true if it worked, false if not. The data that is outed is sent to a variable chosen in the TryParse statement. While the bool that is returned goes back to the place it was called.

Prompt C:

Arrays and lists both store a collection of variables. Arrays are slightly more optimized than lists. An array requires a set length, while a list can store any amount of values. In the case that you knew you only needed two numbers, it would be better to use an array, if you don’t know how many numbers you will get, a list is better. Arrays can also be multidimensional while lists are strictly one-dimensional. This means you can store multiple pieces of data together in a single spot in an array.