Module 2 - Console App Assignment Part 6 - Model Answer

```
using System;
using System.Collections.Generic;
using System.Text;

class Program
{
    static void Main(string[] args)
    {
        // Part 1
        //Create a one - dimensional Array of strings.
        string[] wordstart = {
            "Welcome to The ",
            "I'm Principal, ",
            "Our goal is to provide our students with a "
        };
        string[] wordend = {
            " Academy.",
            ".",
            " learning experience."
        };
        List<string> responses = new List<string>();
        //Ask the user to input some text.
        Console.WriteLine("Please enter a noun:");
        responses.Add(Console.ReadLine());
        Console.WriteLine("Please enter the name of someone infamous:");
        responses.Add(Console.ReadLine());
        Console.WriteLine("Please enter an adjective");
        responses.Add(Console.ReadLine());
        //Create a loop that goes through each string in the Array, adding the user's text to the string.
        for (int i = 0; i < wordstart.Length; i++)
        {
            wordstart[i] += responses[i];
            Console.WriteLine(wordstart[i] + wordend[i]);
        }
        Console.ReadLine();
        Console.WriteLine("The strings we used:");
        //Then create a loop that prints off each string in the Array on a separate line.
        for (int i = 0; i < wordstart.Length; i++)
        {
            Console.WriteLine(wordstart[i] + wordend[i]);
            //Part 2
            // Create an infinite loop (Uncomment line "i--" for infinite loop.
            // Fix the infinite loop.
            // i--;
        }
        Console.ReadLine();
        //Part 3
        Console.WriteLine("Is that a ghost?!");
        StringBuilder boo = new StringBuilder();
        boo.Append("B");
        //Create a loop where the comparison that's used to determine whether to continue iterating the loop is a "<" operator.
        while (boo.Length < 10)
        {
            boo.Append("o");
        }
        boo.Append("!");
        Console.ReadLine();
        Console.WriteLine(boo);
        Console.ReadLine();
        StringBuilder ahh = new StringBuilder();
        ahh.Append("A");
        //Add a loop where the comparison that's used to determine whether to continue iterating the loop is a "<=" operator.
        while (ahh.Length <= 10)
        {
            ahh.Append("h");
        }
        ahh.Append("!");
        Console.WriteLine(ahh);
        Console.ReadLine();
```

```
        //Part 4
        //Create a List of strings where each item in the list is unique.
        List<string> teams = new List<string>() { "Bucks", "Raptors", "Celtics", "Heat", "Pacers", "76ers", "Nets", "Magic",
            "Wizards", "Hornets", "Bulls", "Knicks", "Pistons", "Hawks", "Cavaliers" };
        Console.WriteLine("NBA Eastern Conference Standings");
        //Ask the user to input text to search for in the List.
        Console.WriteLine("Enter team name:");
        int standing = 0;
        bool isValid = false;
        int index = 0;
        //Create a loop that iterates through the list and then displays the index of the array that contains matching text on the screen.
        while (!isValid)
        {
            int i2 = 0;
            string teamrequest = Console.ReadLine();
            foreach (string team in teams)
            {
                if (teamrequest == team)
                {
                    standing = i2 + 1;
                    index = i2;
                }
                i2++;
            }
            //Add code to that above loop that tells a user if they put in text that isn't in the List.
            if (standing == 0)
            {
                Console.WriteLine("That is not a valid team name. Enter again:");
            }
            //Add code to that above loop that stops it from executing once a match has been found.
            else
            {
                isValid = true;
            }
          }
          Console.WriteLine("Processing... Index is: " + index + ". So...");
          Console.WriteLine("Standing is: " + standing);
          Console.ReadLine();

        // Part 5
        // Create a List of strings that has at least two identical strings in the List.
        List<string> tables = new List<string>() { "Reserved", "Vacant", "Taken", "Vacant", "Vacant", "Reserved", "Taken" };
        Console.WriteLine("Welcome to Swell Taco! \nDue to COVID-19 we have a computerized check-in.\nIf you have a reservation type " +
            "\"Reserved\".\nIf you are joining a party that is already here type \"Taken\".\nOtherwise type \"Vacant\".");
        //Ask the user to select text to search for in the List.
        Console.WriteLine("Enter selection:");
        string selection = Console.ReadLine();
        while (!(selection == "Vacant" || selection == "Reserved" || selection == "Taken"))
        {
            //Add code that tells a user if they put in text that isn't in the List.
            Console.WriteLine("Please enter one of the 3 choices:");
            selection = Console.ReadLine();
        }
        Console.WriteLine("The tables that match your selection are numbered:");
        //Create a loop that iterates through the loop and then displays the indices of the array that contain matching text on the screen.
        for (int i3 = 0; i3 < tables.Count; i3++)
        {
            if (selection == tables[i3])
            {
                Console.WriteLine(i3);
            }

        }
        Console.ReadLine();
        //Part 6
        //Create a List of strings that has at least two identical strings in the List.
        List<string> names = new List<string>() { "Bob", "Jim", "Tiffany", "Bob", "Kat", "Mary", "Mary" };
        List<string> repeatCheck = new List<string>();
        Console.WriteLine("Class Roster:");
        //Create a foreach loop that evaluates each item in the list, and displays a message showing the string and whether or not it has already appeared in the list.
        foreach (string name in names)
        {
            Console.WriteLine(name);
            if (repeatCheck.Contains(name))
            {
                Console.WriteLine("This name has been repeated. \nUse last name initial when referring to this student.");
            }
            else
            {
                Console.WriteLine("This name has not been repeated.");
            }
            repeatCheck.Add(name);
        }
        Console.ReadLine();
    }

}
```