```
//Lab Exercise 6.8.2021 Problem 1
//Author: nmessa

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Problem_1
{
   class Program
   {
      static void Main(string[] args)
      {
         //Declare variables
         Random r = new Random();
         int temp;
         int cows= 0;
         int bulls = 0;
         int count;
         int numGuesses = 0;
         string number = "";
         string guess;

         //Build the secret number
         while (number.Length < 4)
         {
            temp = r.Next(10);
            if (!number.Contains(temp.ToString()))
               number += temp.ToString();
         }
         //For testing purposes
         //Console.WriteLine(number);

         //Keep guessing until you get 4 cows
         while (cows < 4)
         {
            //get user guess
            Console.Write("Enter your guess (4 digit number of unique digits): ");
            guess = Console.ReadLine();
            numGuesses++;
            //Count the matches
            count = 0;
            cows = 0;
```

```
            //Find how many numbers in the guess are in the number
            for (int i = 0; i < guess.Length; i++)
            {
                if (number.Contains(guess[i]))
                    count++;
            }

            //Count the cows
            for (int i = 0; i < guess.Length; i++)
            {
                if (number[i] == guess[i])
                    cows++;
            }

            //Calculate the number of bulls
            bulls = count - cows;

            //Output the hint
            Console.WriteLine(cows + " cows " + bulls + " bulls");
        }
        Console.WriteLine("You guessed the number in " + numGuesses + " guesses");
    }
  }
}
```

```
//Lab Exercise 6.8.2021 Problem 2
//Author: nmessa

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Problem_2a
{
   public partial class Form1 : Form
   {

      List<string> countries = new List<string>();
      List<string> capitals = new List<string>();


      public Form1()
      {
         InitializeComponent();
      }

      private void Form1_Load(object sender, EventArgs e)
      {
         //Create FileStream and StreamReader to read capitals.txt
         FileStream fs = new FileStream("capitals.txt", FileMode.Open, FileAccess.Read);
         StreamReader textfile = new StreamReader(fs);

         //Declare local variables
         int listSize = 0;   //Needed to keep track of the size of the lists
         string[] temp;

         //Read first line to prime the loop
         string line = textfile.ReadLine();
```

```
            //Keep looping until you run out of lines of text
            while (line != null)
            {
                //Break line up into country and capital
                temp = line.Split('-');

                //Add country and capital to their respective lists
                countries.Add(temp[0]);
                capitals.Add(temp[1]);
                listSize++;

                //Read in the next line
                line = textfile.ReadLine();
            }
            //Close the StreamReader and FileStream
            textfile.Close();
            fs.Close();

            //Add the countries to the ComboBox items
            for (int i = 0; i < listSize; i++)
            {
                cboCountry.Items.Add(countries[i]);
            }
        }

        //This method will place the capital of the selected country into the label
        private void cboCountry_SelectedIndexChanged(object sender, EventArgs e)
        {
            int index = cboCountry.SelectedIndex;
            lblCapital.Text = capitals[index];
        }


        //This method will exit the application
        private void btnExit_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Thank you for using Capital Finder Pro");
            this.Close();
        }
    }
}
```

```
//Lab Exercise 6.8.2021 Problem 3
//Author: nmessa

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Problem_3
{
   class Program
   {
      static void Main(string[] args)
      {
         string[] suits = new string[]{"clubs", "diamonds", "hearts", "spades"};
         string[] faces = new string[] {"two", "three", "four", "five", "six", "seven", "eight",
            "nine", "ten", "jack", "queen", "king", "ace"};

         string[] deck = new string[52];
         string card;
         int count = 0;

         //Build the deck of cards
         for (int i = 0; i < suits.Length; i++)
         {
            for (int j = 0; j < faces.Length; j++)
            {
               card = faces[j] + " of " + suits[i];
               deck[count] = card;
               count++;
            }
         }

         //print the deck
         for (int i = 0; i < deck.Length; i++)
         {
            Console.WriteLine(deck[i]);
         }
         Console.WriteLine();

         //Shuffle the deck
         shuffle(deck);

         //print the deck after shuffling
         for (int i = 0; i < deck.Length; i++)
         {
            Console.WriteLine(deck[i]);
         }
```

```
//Create and initialize game variables
bool keepGoing = true;
string answer;
int hands = 0;
int ties = 0;
int my_score = 0;
int your_score = 0;
int card_count = 0;
string my_card, your_card, mc, yc;
int myIndex, yourIndex;
string[] mca;
string[] yca;

//Start game loop
while (keepGoing && (hands < 26))
{
   hands += 1;
   my_card = deck[card_count];  //Get card from deck
   card_count += 1;
   your_card = deck[card_count]; //Get card from deck
   card_count += 1;

   //Print each player's hand
   Console.WriteLine("I have the " + my_card);
   Console.WriteLine("You have the " + your_card);
   Console.ReadLine();   //Pause

   //get face value for each card
   mca = my_card.Split(' ');
   mc = mca[0].ToString();
   yca = your_card.Split(' ');
   yc = yca[0].ToString();
```

```
        //Test to see who won
        myIndex = findIndex(faces, mc);
        yourIndex = findIndex(faces, yc);

        if (myIndex > yourIndex)
        {
            Console.WriteLine("I win!");
            my_score += 1 + ties;
            ties = 0;
        }
        else if (myIndex < yourIndex)
        {
            Console.WriteLine("You win!");
            your_score += 1 + ties;
            ties = 0;
        }
        else
        {
            Console.WriteLine("It's a tie!");
            ties += 1;
        }


        //Print current score
        Console.WriteLine("Score: Computer " + my_score + ", You " + your_score);

        //check for another round
        Console.WriteLine("Hit [Enter] to keep going, any other keys to exit: ");
        answer = Console.ReadLine();

        keepGoing = (answer == "");
    }

    //Print results of game
    Console.WriteLine("Game Over");
    if (my_score > your_score)
        Console.WriteLine("I win the game!");
    else if (your_score > my_score)
        Console.WriteLine("You win the game!");
    else
        Console.WriteLine("It was a tie game!");

}
```

```
        static void shuffle(string[] d)
        {
            Random r = new Random();
            int rNumber;
            string temp;
            for (int i = 0; i < d.Length; i++)
            {
                rNumber= r.Next(52);
                temp = d[rNumber];
                d[rNumber] = d[i];
                d[i] = temp;
            }
        }

        static int findIndex(string[] arr, string find)
        {
            int index = 0;
            for (int i = 0; i < arr.Length; i++)
            {
                if (arr[i] == find)
                {
                    index = i;
                    break;
                }
            }
            return index;
        }
    }
}
```

```
//Lab Exercise 6.8.2021 Problem 4
//Author: nmessa

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Problem_4
{
   class Program
   {
      static void Main(string[] args)
      {
         string[] suits = new string[] { "clubs", "diamonds", "hearts", "spades" };
         string[] faces = new string[] {"two", "three", "four", "five", "six", "seven", "eight",
            "nine", "ten", "jack", "queen", "king", "ace"};

         string[] deck = new string[52];
         string card;
         int count = 0;

         //Build the deck of cards
         for (int i = 0; i < suits.Length; i++)
         {
            for (int j = 0; j < faces.Length; j++)
            {
               card = faces[j] + " of " + suits[i];
               deck[count] = card;
               count++;
            }
         }

         //print the deck
         for (int i = 0; i < deck.Length; i++)
         {
            Console.WriteLine(deck[i]);
         }
         Console.WriteLine();

         //Shuffle the deck
         shuffle(deck);
```

```
//print the deck after shuffling
for (int i = 0; i < deck.Length; i++)
{
   Console.WriteLine(deck[i]);
}

//Create and initialize game variables
bool keepGoing = true;
string answer;
int hands = 0;
int ties = 0;
int my_score = 0;
int your_score = 0;
int card_count = 0;
string my_card, your_card, mc, yc;
int myIndex, yourIndex;
string[] mca;
string[] yca;

//Start game loop
while (keepGoing && (hands < 26))
{
   hands += 1;
   my_card = deck[card_count];  //Get card from deck
   card_count += 1;
   your_card = deck[card_count]; //Get card from deck
   card_count += 1;

   //Print each player's hand
   Console.WriteLine("I have the " + my_card);
   Console.WriteLine("You have the " + your_card);
   //Console.ReadLine();   //Pause

   //get face value for each card
   mca = my_card.Split(' ');
   mc = mca[0].ToString();
   yca = your_card.Split(' ');
   yc = yca[0].ToString();


   //Test to see who won
   myIndex = findIndex(faces, mc);
   yourIndex = findIndex(faces, yc);
```

```
        if (myIndex > yourIndex)
        {
            Console.WriteLine("I win!");
            my_score += 1 + ties;
            ties = 0;
        }
        else if (myIndex < yourIndex)
        {
            Console.WriteLine("You win!");
            your_score += 1 + ties;
            ties = 0;
        }
        else
        {
            Console.WriteLine("It's a tie!");
            ties += 1;
        }


        //Print current score
        Console.WriteLine("Score: Computer " + my_score + ", You " + your_score);

        ////check for another round
        //Console.WriteLine("Hit [Enter] to keep going, any other keys to exit: ");
        //answer = Console.ReadLine();

        //keepGoing = (answer == "");
    }

    //Print results of game
    Console.WriteLine("Game Over");
    if (my_score > your_score)
        Console.WriteLine("I win the game!");
    else if (your_score > my_score)
        Console.WriteLine("You win the game!");
    else
        Console.WriteLine("It was a tie game!");
}
```

```
        static void shuffle(string[] d)
        {
            Random r = new Random();
            int rNumber;
            string temp;
            for (int i = 0; i < d.Length; i++)
            {
                rNumber = r.Next(52);
                temp = d[rNumber];
                d[rNumber] = d[i];
                d[i] = temp;
            }
        }

        static int findIndex(string[] arr, string find)
        {
            int index = 0;
            for (int i = 0; i < arr.Length; i++)
            {
                if (arr[i] == find)
                {
                    index = i;
                    break;
                }
            }
            return index;
        }
    }
}
```

```
//Lab Exercise 6.8.2021 Problem 5
//Author: nmessa

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Problem_5
{
   class Program
   {
      static void Main(string[] args)
      {
         //Initialize Variables
         bool keep_going = true;
         int score = 0;
         int count = 0;
         int three = 0;
         int four = 0;
         int five = 0;
         List<int> dice = new List<int>() { 0, 0, 0, 0, 0 };
         Random r = new Random();

         while (keep_going)
         {
            count += 1;

            //roll five dice
            for (int i = 0; i < 5; i++)
               dice[i] = r.Next(1, 7);

            //print result of roll

            printList(dice);

            // Sort the dice
            dice.Sort();

            // Check for five of a kind, four of a kind, three of a kind
            // Yahtzee - all five dice are the same
            if (dice[0] == dice[4])
            {
               Console.WriteLine("Yahtzee!");
               score += 50;
               five += 1;
            }
```

```
            // FourOfAKind - first four are the same, or last four are the same
            else if ((dice[0] == dice[3]) || (dice[1] == dice[4]))
            {
                Console.WriteLine("Four of a kind!");
                score += 25;
                four += 1;
            }

            // ThreeOfAKind - first three, middle three, or last three are the same
            else if ((dice[0] == dice[2]) || (dice[1] == dice[3]) || (dice[2] == dice[4]))
            {
                Console.WriteLine("Three of a kind");
                score += 10;
                three += 1;
            }

            //Keep rolling until you get 1000 points
            if (score >= 1000)
                keep_going = false;
        }//end of while loop

        //print result of game
        Console.WriteLine("It took " + count + " rolls to get 1000 points");
        Console.WriteLine("Threes: " + three);
        Console.WriteLine("Fours: " + four);
        Console.WriteLine("Yahtzee: " + five);
    }

    static void printList(List<int> c)
    {
        Console.Write("You rolled: ");
        for (int i = 0; i < 5; i++)
            Console.Write(c[i] + " ");
        Console.WriteLine();
    }
  }
}
```