```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO.Pipes;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace Homework02
{
    public class Card
    {
        public string Value { get; set; }
        public int Point { get; set; }

        public Card(string value, int point)
        {
            Value = value;
            Point = point;
        }
    }
    internal class Program
    {
        static void Terminate(string isTerminate)
        {
            isTerminate = isTerminate.ToLower();
            if (isTerminate == "exit")
            {
```

```
            Console.WriteLine("\nThe program terminates. Thank you for using my program.");
            Console.WriteLine("We are looking forward to seeing you again.");
            Console.ReadKey();
            Environment.Exit(0);
        }
    }
    static void Main(string[] args)
    {
        string program = "";
        string temporary = "";
        do
        {
            Console.Write("Which Program do you wan to start? (1 = Signum, 2 = Divide, 3 = 21, 4 =
LNKO, Exit\nYou can use the Exit in any input: ");
            program = Console.ReadLine();
            Terminate(program);
            if (((program != "1" && program != "2") && (program != "3" && program != "exit")) && program
!= "4")
            {
                Console.WriteLine("Invalid input\n");
                continue;
            }
            bool invalidInput = false;
            switch (program)
            {
                case "1":
                    do
                    {
                        invalidInput = false;
                        Console.Write("Enter a number: ");
```

```
double numberToSignum;
temporary = Console.ReadLine();
int signum;
Terminate(temporary);
if (!double.TryParse(temporary, out numberToSignum))
{
  Console.WriteLine("Invalid input\n");
  invalidInput = true;
  continue;
}
if (numberToSignum < 0)
{
  signum = -1;
}
else if (numberToSignum > 0)
{
  signum = 1;
}
else
{
  signum = 0;
}

Console.Write("The value of the signum based on your number ({0}) is \"{1}\" (",
numberToSignum, signum);
if (signum == -1)
{
  Console.Write("-).\n\n");
}
```

```
        else
        {
            Console.Write("+).\n\n");
        }
    } while (invalidInput);
    break;

case "2":
    int number1; // >= 0 - input1
    int number2; // > 0 - input2 && < input1
    int result = 0; // >= 0 - output

    do
    {
        invalidInput = false;
        // Input1
        Console.Write("Enter the first number: ");
        temporary = Console.ReadLine();
        // Checking the input
        Terminate(temporary);
        if (!int.TryParse(temporary, out number1) || number1 < 0)
        {
            Console.WriteLine("Invalid input\n");
            invalidInput = true;
            continue;
        }

        // Input2
        Console.Write("Enter the second number: ");
```

```
            temporary = Console.ReadLine();
            // Checking the input
            Terminate(temporary);
            if ((!int.TryParse(temporary, out number2) || number2 <= 0) || number2 > number1)
            {
                Console.WriteLine("Invalid input\n");
                invalidInput = true;
                continue;
            }

            // Calculation
            int temp = number1;
            while (temp >= number2)
            {
                temp -= number2;
            }
            result = temp;
        }
        while (invalidInput);

        Console.WriteLine("The remainings of the division is: {0}\n", result);

        break;

    case "3":
        List<Card> cards = new List<Card>();
        cards.Add(new Card("alsó", 2));
        cards.Add(new Card("felső", 3));
        cards.Add(new Card("király", 4));
```

```
cards.Add(new Card("vii", 7));
cards.Add(new Card("viii", 8));
cards.Add(new Card("ix", 9));
cards.Add(new Card("x", 10));
cards.Add(new Card("ász", 11));

bool isCard = false;
string cardName = "";

do
{
   Console.Write("Please enter a card name: ");
   temporary = Console.ReadLine();
   Terminate(temporary);
   cardName = temporary;
   cardName = cardName.ToLower();

   for (int i = 0; i < cards.Count; i++)
   {
      if (cardName == cards[i].Value)
      {
         isCard = true;
         Console.WriteLine("\nThe value of the {0} is: {1}\n", cards[i].Value, cards[i].Point);
         break;
      }
   }
   if (!isCard)
   {
```

```
                Console.WriteLine("\nInvalid card. You entered \"{0}\". Please review your input and enter a valid card.\n", cardName);
            }
        } while (!isCard);

        break;

    case "4":
        int number1_LNKO;
        int number2_LNKO;
        do
        {
            invalidInput = false;
            //First number
            Console.Write("Please enter the first number: ");
            temporary = Console.ReadLine();
            Terminate(temporary);
            if (!int.TryParse(temporary, out number1_LNKO) || number1_LNKO <= 0)
            {
                Console.WriteLine("Invalid input.\n");
                invalidInput = true;
                continue;
            }

            //Second number
            Console.Write("Please enter the second number: ");
            temporary = Console.ReadLine();
            Terminate(temporary);
            if (!int.TryParse(temporary, out number2_LNKO) || number2_LNKO <= 0)
```

```
            {
                Console.WriteLine("\nInvalid input.\n");
                invalidInput = true;
                continue;
            }

            //Calculation
            if (number1_LNKO < number2_LNKO)
            {
                int temp = number1_LNKO;
                number1_LNKO = number2_LNKO;
                number2_LNKO = temp;
            }

            int remainder = number1_LNKO % number2_LNKO;
            while (remainder > 0)
            {
                number1_LNKO = number2_LNKO;
                number2_LNKO = remainder;
                remainder = number1_LNKO % number2_LNKO;
            }
            Console.WriteLine("\nThe Greatest Common Divisor is: {0}\n", number2_LNKO);

        } while (invalidInput);
        break;
    }
  }
  while (program != "exit");
}
```

```
  }
}
```