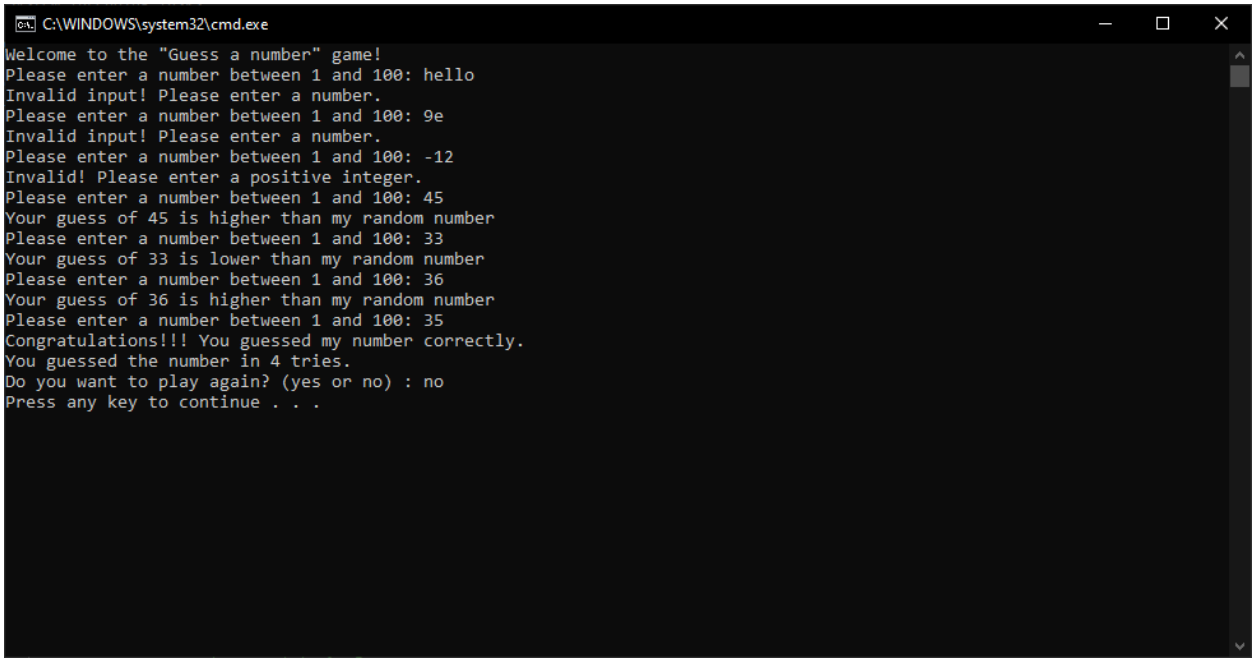


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HW_MIDTERM
{
   class Program
   {
      /* Psuedo code
      For this game we need random number, a user input and a comparison.
      So we can split the code into GetInput, random_num and H_or_L functions.
      GetInput validates the user inputs; repeats if not valid; and returns the number.
      random_num creates a random number on comand.
      The H_or_L simply writes a response to the user inputs.
      */
      /* Code flow
      main while loop to create multiple matches
      input validation and loop
      win condition
      repeat match function
      */
      static void Main(string[] args)
      {
```

```
    // Declarations
    bool win = false;
    string user_says = "";
    int tries = 0;
    int random = random_num(1, 101);
    // Welcome
    Console.WriteLine("Welcome to the \"Guess a number\" game!");
    // Debugging
    // Console.WriteLine(random);
    // Main while loop to keep playing
    while (true)
    {
        // Game inputs only happens when user hasn't won
        if (!win)
        {
            int input = GetInput(1, 100, "Please enter a number between 1 and 100: ");
            win = Num_check(input, random, ref tries);
        }
        // When user wins another set of inputs is required
        if (win)
        {
            Console.Write("Do you want to play again? (yes or no) : ");
            user_says = Console.ReadLine().ToLower();
            // Conditions for game continuation
            if (user_says == "no")
            {
                break;
            }
            else if (user_says == "yes")
            {
                // new random number is created and resets the game
                random = random_num(1, 101);
                win = false;
                tries = 0;
            }
            else
            {
                Console.WriteLine("Please input yes or no!");
            }
        }
    }
}

// Input validation and error loop function
```

```
static int GetInput(int low, int high, string to_say)
{
    int y;
    bool valid = false;
    bool in_range = false;
    do
    {
        Console.Write(to_say);
        valid = int.TryParse(Console.ReadLine(), out y);
        in_range = (y > low) && (y < high);
        if (!valid)
        {
            Console.WriteLine("Invalid input! Please enter a number.");
        }
        else if (!in_range)
        {
            Console.WriteLine("Invalid! Please enter a positive integer.");
        }
    }
    while ((!valid) || (!in_range));
    return y;
}

// Generates a random number between limits
static int random_num(int low, int high)
{
    Random rnd = new Random();
    int random = rnd.Next(low, high);
    return random;
}

// Checks the input with random number and writes to console
// Also handles the number of tries counter
static bool Num_check(int input, int random, ref int tries)
{
    bool num_check = false;
    if (input > random)
    {
        Console.WriteLine($"Your guess of {input} is higher than my random number");
        tries++;
    }
    else if (input < random)
    {
        Console.WriteLine($"Your guess of {input} is lower than my random number");
```

```
            tries++;
        }
        else if (input == random)
        {
            tries++;
            Console.WriteLine("Congratulations!!! You guessed my number correctly.");
            Console.WriteLine($"You guessed the number in {tries} tries.");
            num_check = true;
        }
        return num_check;
    }

  }
}
```