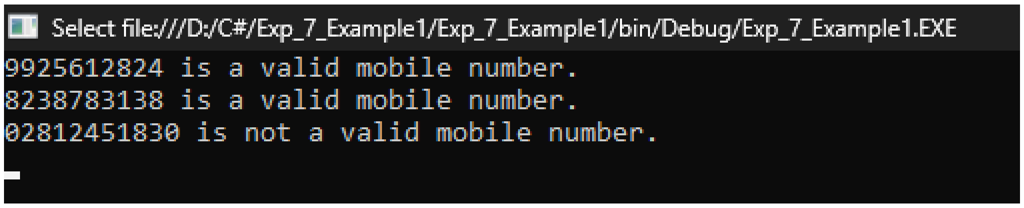
Example 1: Below example demonstrate the use of regex in Mobile Number Verification. Suppose you are making a form where you need to verify the user-entered mobile number then you can use regex.

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

namespace Exp_7_Example1
{
    class Program
    {
        static void Main(string[] args)
        {
            // Input strings to Match
            // valid mobile number
            string[] str = { "9925612824",
                             "8238783138",
                             "02812451830" };

            foreach (string s in str)
            {
                Console.WriteLine("{0} {1} a valid mobile number.", s,
                                  isValidMobileNumber(s) ? "is" : "is
not");
            }

            Console.ReadKey();
        }

        // Method containing the regex (fixed indentation)
        public static bool isValidMobileNumber(string
inputMobileNumber)
        {
            string strRegex = @"(^[0-9]{10}$)|(^\+[0-9]{2}\s+[0-
9]{2}[0-9]{8}$)|(^[0-9]{3}-[0-9]{4}-[0-9]{4}$)";

            Regex re = new Regex(strRegex);

            if (re.IsMatch(inputMobileNumber))
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    }
}
```

Output:



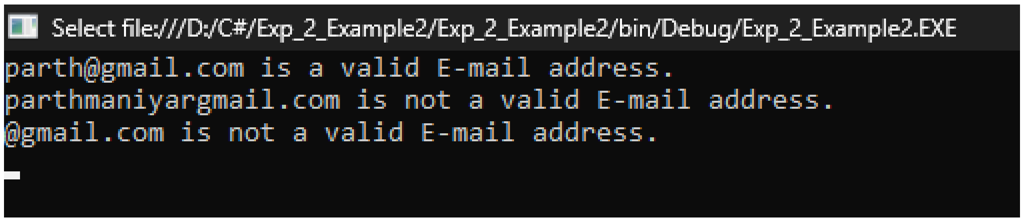
```
Select file:///D:/C#/Exp_7_Example1/Exp_7_Example1/bin/Debug/Exp_7_Example1.EXE
9925612824 is a valid mobile number.
8238783138 is a valid mobile number.
02812451830 is not a valid mobile number.
```

Example 2: Below example demonstrate the use of regex in Email ID Verification. Suppose you are making a form where you need to verify the user-entered email id then you can use regex.

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

namespace Exp_2_Example2
{
    class Program
    {
        static void Main(string[] args)
        {
            // Input strings for Match
            // valid E-mail address.
            string[] str = {"parth@gmail.com",
                  "parthmaniyargmail.com",
                            "@gmail.com"};

            foreach (string s in str)
            {
                Console.WriteLine("{0} {1} a valid E-mail address.",
s,
                                    isValidEmail(s) ? "is" : "is
not");
```

```
            }
            Console.ReadKey();
        }

        // Method to check the Email ID
        public static bool isValidEmail(string inputEmail)
        {

            // This Pattern is use to verify the email
            string strRegex = @"\A(?:[a-z0-9!#$%&'*+/=?^_`{|}~-
]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-
9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?)\Z";

            Regex re = new Regex(strRegex, RegexOptions.IgnoreCase);

            if (re.IsMatch(inputEmail))
                return (true);
            else
                return (false);
            Console.ReadKey();
        }
    }
}
```

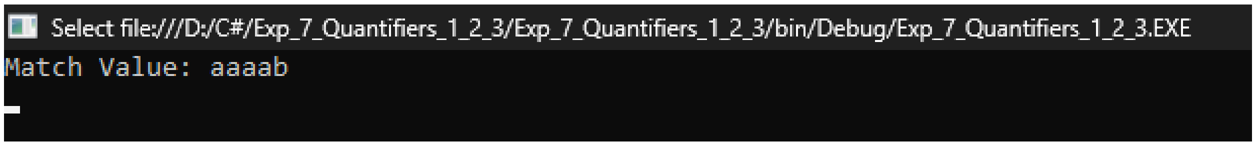
Output:


Select file:///D:/C#/Exp_2_Example2/Exp_2_Example2/bin/Debug/Exp_2_Example2.EXE

```
parth@gmail.com is a valid E-mail address.
parthmaniyargmail.com is not a valid E-mail address.
@gmail.com is not a valid E-mail address.
```

Quantifiers:

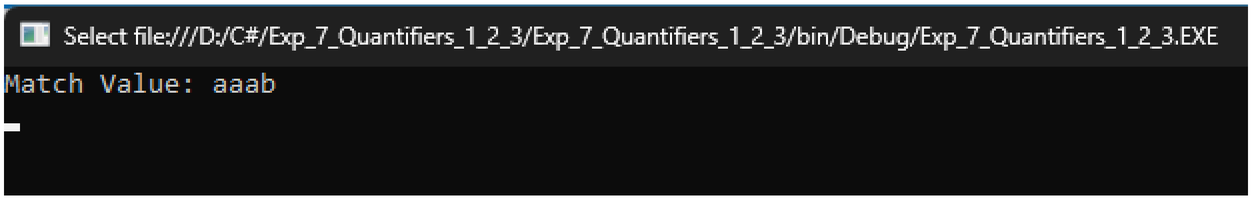
Example 1:
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;

namespace Exp_7_Quantifiers_1_2_3
{
    class Program
    {
        static void Main(string[] args)
        {

        // pattern b, ab, aab, ...
        Regex regex = new Regex(@"a*b");

        Match match = regex.Match("aaaabcd");
        if (match.Success)
        {
            Console.WriteLine("Match Value: " + match.Value);
        }
        Console.ReadKey();

        }
    }
}
```
Output:


```
Select file:///D:/C#/Exp_7_Quantifiers_1_2_3/Exp_7_Quantifiers_1_2_3/bin/Debug/Exp_7_Quantifiers_1_2_3.EXE
Match Value: aaaab
```

Example 2:
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
```

```
namespace Exp_7_Quantifiers_1_2_3
{
    class Program
    {
        static void Main(string[] args)
        {

            // this will return any pattern
            // like ab, aab, aaab, ....
            Regex regex = new Regex(@"a+b");
            Match match = regex.Match("aaabcd");
            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }
            Console.ReadKey();

        }
    }
}
```

Output:

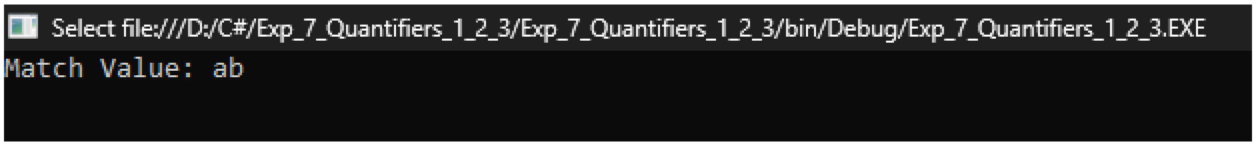


```
Select file:///D:/C#/Exp_7_Quantifiers_1_2_3/Exp_7_Quantifiers_1_2_3/bin/Debug/Exp_7_Quantifiers_1_2_3.EXE
Match Value: aaab
```

Example 3:

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

namespace Exp_7_Quantifiers_1_2_3
{
    class Program
    {
        static void Main(string[] args)
        {
            // This return any pattern like b, ab
            Regex regex = new Regex(@"a?b");

            Match match = regex.Match("aaaabcd");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }

            Console.ReadKey();

        }
    }
}
```

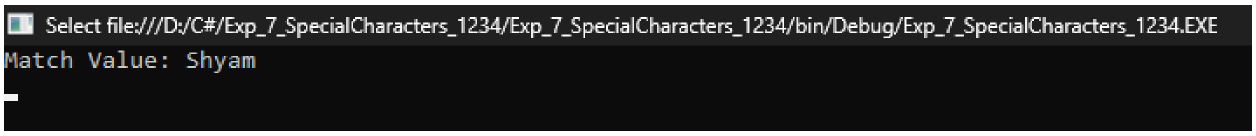
Output:



```
Select file:///D:/C#/Exp_7_Quantifiers_1_2_3/Exp_7_Quantifiers_1_2_3/bin/Debug/Exp_7_Quantifiers_1_2_3.EXE
Match Value: ab
```

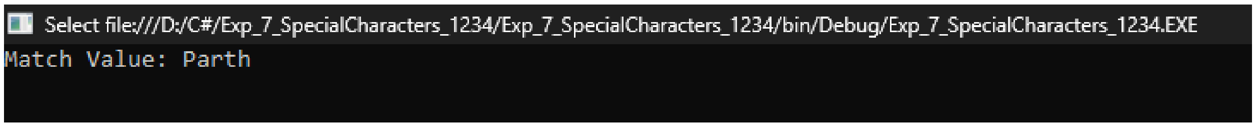
Special Characters:

Example 1:
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;

namespace Exp_7_SpecialCharacters_1234
{
    class Program
    {
        static void Main(string[] args)
        {
            // This will return if shyam exist
            // at the beginning of the line
            Regex regex = new Regex(@"^Shyam");

            Match match = regex.Match("Shyam is my pet name");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }
            Console.ReadKey();
        }
    }
}
```
Output:


Select file:///D:/C#/Exp_7_SpecialCharacters_1234/Exp_7_SpecialCharacters_1234/bin/Debug/Exp_7_SpecialCharacters_1234.EXE
```
Match Value: Shyam
```

Example 2:
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
```

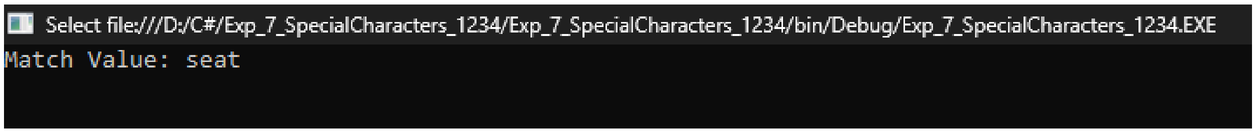
```
namespace Exp_7_SpecialCharacters_1234
{
    class Program
    {
        static void Main(string[] args)
        {

            // This return parth if it
            // exist at the end of the line
            Regex regex = new Regex(@"Parth$");

            Match match = regex.Match("My name is Parth");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }

            Console.ReadKey();
        }
    }
}
```
Output:



```
Select file:///D:/C#/Exp_7_SpecialCharacters_1234/Exp_7_SpecialCharacters_1234/bin/Debug/Exp_7_SpecialCharacters_1234.EXE
Match Value: Parth
```

Example 3:
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;

namespace Exp_7_SpecialCharacters_1234
{
    class Program
    {
        static void Main(string[] args)
        {
            // This will return any word which
            // contains only one letter between
            // s and t
            Regex regex = new Regex(@"s..t");
            Match match = regex.Match("This is my seat");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }

            Console.ReadKey();
        }
    }
}
```

Output:

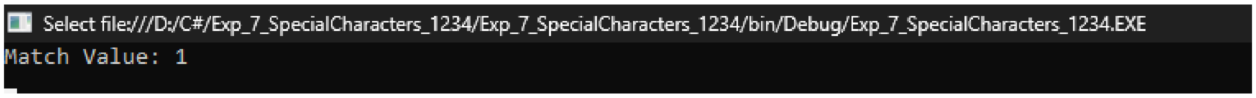


```
Select file:///D:/C#/Exp_7_SpecialCharacters_1234/Exp_7_SpecialCharacters_1234/bin/Debug/Exp_7_SpecialCharacters_1234.EXE
Match Value: seat
```

Example 4:

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

namespace Exp_7_SpecialCharacters_1234
{
    class Program
    {
        static void Main(string[] args)
        {
            // This will the return
            // the one digit character
            Regex regex = new Regex(@"\d");

            Match match = regex.Match("I am 19 years old");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }
            Console.ReadKey();
        }
    }
}
```

Output:



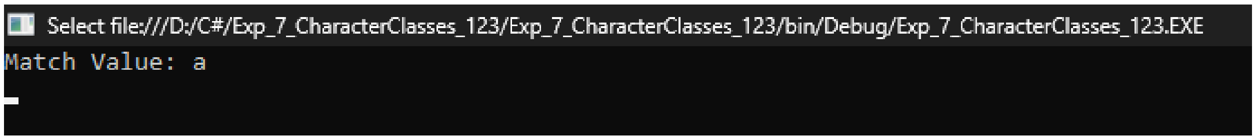
```
Select file:///D:/C#/Exp_7_SpecialCharacters_1234/Exp_7_SpecialCharacters_1234/bin/Debug/Exp_7_SpecialCharacters_1234.EXE
Match Value: 1
```

Character Classes:

Example 1:

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

namespace Exp_7_CharacterClasses_123
{
    class Program
    {
        static void Main(string[] args)
        {
            // This will return one character either
            // a or b or c which will come first
            Regex regex = new Regex(@"[abc]");

            Match match = regex.Match("abcdef");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }
            Console.ReadKey();
        }
    }
}
```

Output:

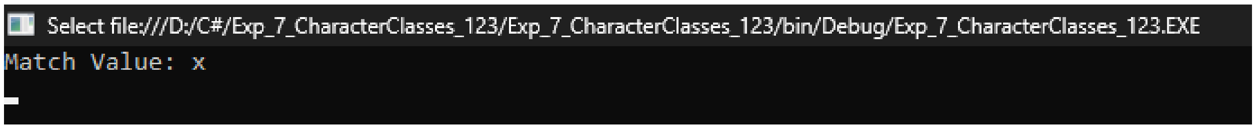


```
Select file:///D:/C#/Exp_7_CharacterClasses_123/Exp_7_CharacterClasses_123/bin/Debug/Exp_7_CharacterClasses_123.EXE
Match Value: a
```

Example 2:

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

namespace Exp_7_CharacterClasses_123
{
    class Program
    {
        static void Main(string[] args)
        {

            // This will return any character
            // between x and z inclusive
            Regex regex = new Regex(@"[x-z]");

            Match match = regex.Match("xmax");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }

            Console.ReadKey();
        }
    }
}
```

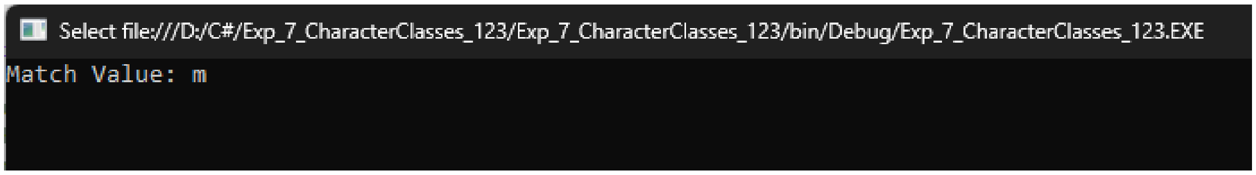
Output:



```
Select file:///D:/C#/Exp_7_CharacterClasses_123/Exp_7_CharacterClasses_123/bin/Debug/Exp_7_CharacterClasses_123.EXE
Match Value: x
```

Example 3:

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

namespace Exp_7_CharacterClasses_123
{
    class Program
    {
        static void Main(string[] args)
        {

            // This will return other x,
            // y and z character
            Regex regex = new Regex(@"[^x-z]");

            Match match = regex.Match("xmax");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }


            Console.ReadKey();
        }
    }
}
```
Output:



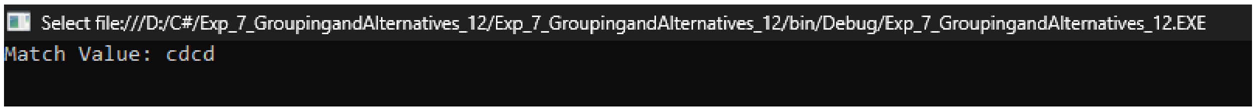
```
Select file:///D:/C#/Exp_7_CharacterClasses_123/Exp_7_CharacterClasses_123/bin/Debug/Exp_7_CharacterClasses_123.EXE
Match Value: m
```

Grouping and Alternatives

Example 1:

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

namespace Exp_7_GroupingandAlternatives_12
{
    class Program
    {
        static void Main(string[] args)
        {
            // This will return pattern
            // will cd, cdcd, cdcdcd, ...
            Regex regex = new Regex(@"(cd)+");

            Match match = regex.Match("cdcdde");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }
            Console.ReadKey();
        }
    }
}
```

Output:



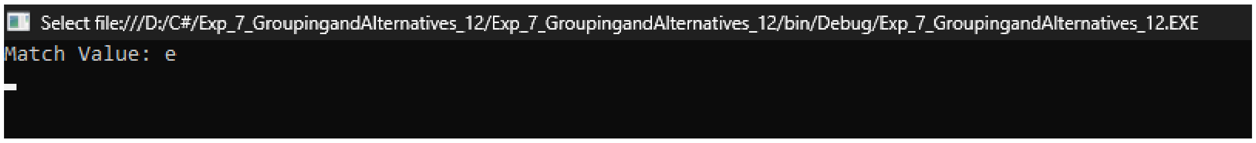
```
Select file:///D:/C#/Exp_7_GroupingandAlternatives_12/Exp_7_GroupingandAlternatives_12/bin/Debug/Exp_7_GroupingandAlternatives_12.EXE
Match Value: cdcd
```

Example 1:

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

```
namespace Exp_7_GroupingandAlternatives_12
{
    class Program
    {
        static void Main(string[] args)
        {
            // This will either d or e
            // which ever comes first
            Regex regex = new Regex(@"d|e");

            Match match = regex.Match("edge");

            if (match.Success)
            {
                Console.WriteLine("Match Value: " + match.Value);
            }

            Console.ReadKey();
        }
    }
}
```

Output:



Select file:///D:/C#/Exp_7_GroupingandAlternatives_12/Exp_7_GroupingandAlternatives_12/bin/Debug/Exp_7_GroupingandAlternatives_12.EXE

```
Match Value: e
```

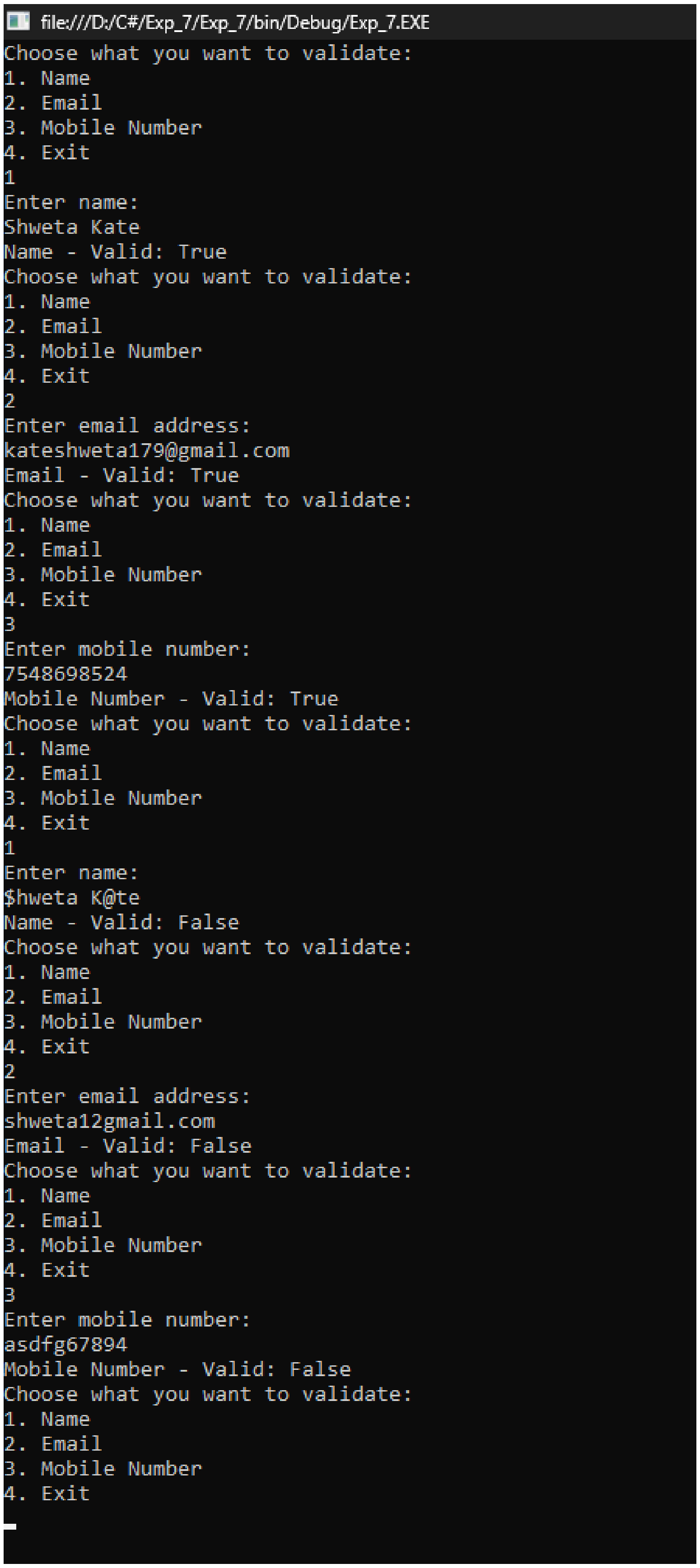
Name: Kate Shweta Sanjay
Roll No.: 3083
Div: B Batch: T4

Problem Statement:
Write Program to validate following data:
EmailID, Mobile No. and Name.

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

namespace Exp_7
{
    class Program
    {

static void Main(string[] args)
        {
            // Regular expressions for mobile number, email address,
and name
            string mobileRegex = @"^\d{10}$";
            string emailRegex = @"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-
]+\.[a-zA-Z]{2,}$";
            string nameRegex = @"^[a-zA-Z\s]+$";

            bool exitRequested = false;

            while (!exitRequested)
            {
                Console.WriteLine("Choose what you want to
validate:");
                Console.WriteLine("1. Name");
                Console.WriteLine("2. Email");
                Console.WriteLine("3. Mobile Number");
                Console.WriteLine("4. Exit");

                int choice;
                if (!int.TryParse(Console.ReadLine(), out choice) ||
choice < 1 || choice > 4)
                {
                    Console.WriteLine("Invalid choice. Please choose a
number between 1 and 4.");
```

```
                    continue;
                }

                string userInput;
                switch (choice)
                {
                    case 1:
                        Console.WriteLine("Enter name:");
                        userInput = Console.ReadLine();
                        Console.WriteLine("Name - Valid: " +
Regex.IsMatch(userInput, nameRegex));
                        break;
                    case 2:
                        Console.WriteLine("Enter email address:");
                        userInput = Console.ReadLine();
                        Console.WriteLine("Email - Valid: " +
Regex.IsMatch(userInput, emailRegex));
                        break;
                    case 3:
                        Console.WriteLine("Enter mobile number:");
                        userInput = Console.ReadLine();
                        Console.WriteLine("Mobile Number - Valid: " +
Regex.IsMatch(userInput, mobileRegex));
                        break;
                    case 4:
                        exitRequested = true;
                        break;
                }
            }

            Console.WriteLine("Exiting program...");
        }


    }
}
```

Output:



file:///D:/C#/Exp_7/Exp_7/bin/Debug/Exp_7.EXE

```
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
1
Enter name:
Shweta Kate
Name - Valid: True
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
2
Enter email address:
kateshweta179@gmail.com
Email - Valid: True
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
3
Enter mobile number:
7548698524
Mobile Number - Valid: True
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
1
Enter name:
$hweta K@te
Name - Valid: False
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
2
Enter email address:
shweta12gmail.com
Email - Valid: False
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
3
Enter mobile number:
asdfg67894
Mobile Number - Valid: False
Choose what you want to validate:
1. Name
2. Email
3. Mobile Number
4. Exit
```