به نام خدا

تمرین سری اول درس کامپایلر

بهاره کاوسی نژاد – 99431217

1 – با استفاده از Regex در زبان C# یا پایتون موارد زیر را پیاده سازی نمایید.

الف – فرمت ایمیل

```
using System;
using System.Text.RegularExpressions;

namespace Compiler
{
    class HW1
    {
        static bool IsEmailValid (string EmailAddress)
        {
            string EmailPattern = @"^[a-zA-Z0-9_.+-]+@[a-zA-Z0-
9-]+\.[a-zA-Z0-9-.]+$";
            Regex regex = new Regex(EmailPattern);
            Match match = regex.Match(EmailAddress);
            return match.Success;
        }
        static void Main ()
        {
            string email = Console.ReadLine();
            if (IsEmailValid(email))
            {
                Console.WriteLine("Email is valid.");
            }
            else
            {
                Console.WriteLine("Email is not valid");
            }
        }
    }
}
```

در EmailPattern، Regex فرمت ایمیل مشخص شده است؛ هر بخش رشته ورودی را به شکل زیر چک می کند:

- ^: شروع رشته
- [a-zA-Z0-9_.+-]+: تعداد یکی یا بیشتر از حروف کوچک و بزرگ انگلیسی، ارقام و علامت های underscore، نقطه، جمع و hyphen
- @: علامت @ در ایمیل
- [a-zA-Z0-9-.]+: تعداد یکی یا بیشتر از حروف کوچک و بزرگ انگلیسی، ارقام و علامت hyphen
- .\: علامت نقطه
- $: انتهای رشته

```
Hello23_@gmail23.com
Email is valid.

34World
Email is not valid
```

ب – عبارت ریاضی (شامل عملیات های ضرب، تقسیم، جمع و تفریق)

```
using System;
using System.Text.RegularExpressions;

namespace Compiler
{
    class HW1
    {
        static bool IsValidMathExpression (string
MathExpression)
        {
            string MathExpressionPattern = @"^\d+(\s*[-
+*/]\s*\d+)*$";
            Regex regex = new Regex(MathExpressionPattern);
            Match match = regex.Match(MathExpression);
            return match.Success;
        }

        static void Main ()
        {
            string MathExpression = Console.ReadLine();
            if (IsValidMathExpression(MathExpression))
            {
                Console.WriteLine("Math expression is valid.");
            }
            else
            {
                Console.WriteLine("Math expression is not
valid");
            }
        }
    }
}
```

در MathExpressionPattern، Regex فرمت عبارت ریاضی مشخص شده است؛ هر بخش رشته ورودی را به شکل زیر چک می کند:

- ^: شروع رشته

- +\d: تعداد یکی یا بیشتر از ارقام
- *(+\d*\s*\s*[-+*/]*\s): تکرار یکی از عملیات های جمع، تقریق، ضرب یا تقسیم که یک عدد به دنبال آن بیاید:
  - *\s: تعداد صفر یا بیشتر space قبل یا بعد از علامت
  - [-+*/]: یکی از عملیات های تفریق، جمع، ضرب یا تقسیم
  - *\s: تعداد صفر یا بیشتر space قبل یا بعد از رقم
  - +\d: تعداد یک یا بیشتر رقم
- *: تکرار گروه قبلی به تعداد صفر یا بیشتر
- $: انتهای رشته

```
25 * 45 - 76 / 2
Math expression is valid.

23
Math expression is valid.

ereds - 24
Math expression is not valid
```

ج – URL

```
using System;
using System.Text.RegularExpressions;

namespace Compiler
{
    class HW1
    {
        static bool IsValidURL (string URL)
        {
            string URLPattern = @"^(https?|ftp)://[^\s/$.?#].
[^\s]*$";
            Regex regex = new Regex(URLPattern);
            Match match = regex.Match(URL);
            return match.Success;
        }

        static void Main ()
        {
            string URL = Console.ReadLine();
            if (IsValidURL(URL))
            {
                Console.WriteLine("URL is valid.");
            }
            else
            {
                Console.WriteLine("URL is not valid");
            }
        }
    }
}
```

در URLPattern، Regex فرمت عبارت ریاضی مشخص شده است؛ هر بخش رشته ورودی را به شکل زیر چک می کند:

- ^: شروع رشته
- (https?ftp): آمدن یکی از رشته های http، https یا ftp. علامت ؟، s را optional می کند تا هم https ممکن باشد و هم http.

- //: : آمدن //:
- [^\s/$.?#]: آمدن هر رشته ای که whitespace، forward slash، dollar sign، نقطه، علامت سوال و یا hash symbol نباشد.
- .: آمدن هر کاراکتری
- *[^\s]: تکرار صفر یا بیشتر از هر کاراکتری که whitespace نباشد.
- $: انتهای رشته

```
https://hello.com
URL is valid.

hello.com
URL is not valid
```

د – کد پستی

```
using System;
using System.Text.RegularExpressions;

namespace Compiler
{
    class HW1
    {
        static bool IsValidPostalCode (string PostalCode)
        {
            string PostalCodePattern = @"^\d{10}$";
            Regex regex = new Regex(PostalCodePattern);
            Match match = regex.Match(PostalCode);
            return match.Success;
        }

        static void Main ()
        {
            string PostalCode = Console.ReadLine();
            if (IsValidPostalCode(PostalCode))
            {
                Console.WriteLine("PostalCode is valid.");
            }
            else
            {
                Console.WriteLine("PostalCode is not valid");
            }
        }
    }
}
```

در PostalCodePattern، Regex فرمت عبارت ریاضی مشخص شده است؛ هر بخش رشته ورودی را به شکل زیر چک می کند:

- ^: شروع رشته
- \d{10}: دقیقا 10 رقم
- $: انتهای رشته

```
1234567891
PostalCode is valid.

fdse456po7
PostalCode is not valid

123456
PostalCode is not valid
```

ه – شماره تلفن

```
using System;
using System.Text.RegularExpressions;

namespace Compiler
{
    class HW1
    {
        static bool IsValidPhoneNumber (string PhoneNumber)
        {
            string PhoneNumberPattern = @"^09\d{9}$";
            Regex regex = new Regex(PhoneNumberPattern);
            Match match = regex.Match(PhoneNumber);
            return match.Success;
        }

        static void Main ()
        {
            string PhoneNumber = Console.ReadLine();
            if (IsValidPhoneNumber(PhoneNumber))
            {
                Console.WriteLine("Phone Number is valid.");
            }
            else
            {
                Console.WriteLine("Phone Number is not valid");
            }
        }
    }
}
```

در PhoneNumberPattern، Regex فرمت عبارت ریاضی مشخص شده است؛ هر بخش رشته ورودی را به شکل زیر چک می کند:

- ^: شروع رشته
- 09: 09 ابتدای شماره تلفن را مشخص می کند
- \d{9}: دقیقا 9 رقم دیگر داریم

- $: انتهای رشته

```
091234567891
Phone Number is not valid

09123456789
Phone Number is valid.

12345678910
Phone Number is not valid

ffdr5671256
Phone Number is not valid
```