# Answer on Question #48223 – Engineering – Other

## Write a function Boolean isValidPassword (String password)

A Password is valid if it satisfies following conditions:
1. Length of the password must be more than 10(without counting spaces).
2. Password must contain at least one capital and one small alphabet
3. Password must contain at least 2 digits and if it does not have 2
digits then see next condition.
4. Password must contain at least one special character from the given set- {$, @, _, -,., /}

For Ex:

Password: CraterZone Info@tech (Valid)
Password: CraterZone 2Info3tech (Valid)
Password: CraterZone Infotech (Invalid since it does not have any special symbol and does not have 2 digits also)
Password: Craterzone@ (Invalid since length of the password is not more than 10)

## Solution (C#)

**Variant 1**

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

public static bool IsValidPassword(string password)
    {
        // array of string for regular expressions with seeking parameters
        string[] pattern = new string[4];
        pattern[0] = @"[a-z]"; // string for searching small alphabits
        pattern[1] = @"[A-Z]"; // string for searching capital alphabits
        pattern[2] = @"[0-9]"; // string for searching digits
        pattern[3] = @"[$@_.-/]"; // string for searching symbols
        // creating regular expression objects
        System.Text.RegularExpressions.Regex[] rgx = new Regex[pattern.Length];
        // initialize regular expression for every pattern strings
        for (int i=0; i<pattern.Length; i++)
            rgx[i] = new Regex(pattern[i]);

        int count = 0; // counter of characters number
        // counting the number of characters in the password without gaps
        foreach(char ch in password)
        {
            if (ch == ' ') continue;
            else count++;
        }
```

```
        // validating that character number more then 10
        if (count <= 10)
        {
           Console.WriteLine("Invalid since length of the password is not more than 10");
           Console.ReadKey();
           return false;
        }
        // validating: one small and one capital are present
        if ((rgx[0].Match(password).Success == true) && (rgx[1].Match(password).Success == true))
        {
           // validating: there are 2 digits or one special symbol at least in the password
           if ((rgx[2].Matches(password).Count >= 2) || (rgx[3].Match(password).Success == true))
           {
              Console.WriteLine("Valid");
              return true;
           }
           else Console.WriteLine("The count of digits is less then two OR there are not any symbols
of: $, @, _, ., -, /");
        }
        // writing what is the reason if password isn't correct
        else if (rgx[0].Match(password).Success == false)
           {
              Console.WriteLine("There isn't any small alphabet");
              return false;
           }
           else if (rgx[1].Match(password).Success == false)
           {
              Console.WriteLine("There isn't any capital alphabet");
              return false;
           }
        return false;
     }
```

**Variant 2**

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

public static bool IsValidPassword(string password)
     {
        int small = 0;
        int cap = 0;
        int dig = 0;
        int symb = 0;

        int count = 0; // counter of characters number
        // counting the number of characters in the password without gaps
```

```
        foreach (char ch in password)
        {
            if (ch == ' ') continue;
            else count++;
            if (ch >= 'a' && ch <= 'z') small++;           // counting small alphabits
            else if (ch >= 'A' && ch <= 'Z') cap++;        // counting capital alphabits
            else if (ch >= '0' && ch <= '9') dig++;        // counting digits
            else if (ch == '$' || ch == '@' || ch == '_' || ch == '.' || ch == '-' || ch == '/') symb++; //
counting symbols
        }

        // validating that character number more then 10
        if (count <= 10)
        {
            Console.WriteLine("Invalid since length of the password is not more than 10");
            Console.ReadKey();
            return false;
        }
        // validating: one small and one capital are present
        if ( (small>0) && (cap>0) )
        {
            // validating: there are 2 digits or one special symbol at least in the password
            if ( (dig >= 2) || (symb>0))
            {
                Console.WriteLine("Valid");
                return true;
            }
            else Console.WriteLine("The count of digits is less then two OR there are not any symbols
of: $, @, _, ., -, /");
        }
        // writing what is the reason if password isn't correct
        else if (small == 0)
        {
            Console.WriteLine("There isn't any small alphabet");
            return false;
        }
        else if (cap == 0)
        {
            Console.WriteLine("There isn't any capital alphabet");
            return false;
        }
        return false;
    }
```