

Forum:[Hack Forums](#)

Thread: [\[C#\] The real Brute Force by Mathian](#)

Mathian - December 26, 2018, 11:16 am

Hi! When I first heard about BruteForce, I imagined going over all the options from "a" to "z". I have created such code long ago, but it would be worth thinking about exceptions:

```
Code using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
```

```
namespace Cycle
{
    class Program
    {
        static void Main(string[] args)
        {
            string Chars = "Alphabet";
            for (int i = 0; i <= 3; i++)
            {
                foreach (var Password in GetCombinations(Chars.ToArray(), i))
                    Console.WriteLine(Password);
            }
            Console.ReadLine();
        }
        private static IEnumerable<string> GetCombinations(char[] Chars, int MaximumLength)
        {
            if (MaximumLength < 1) yield break;
            foreach (var a in Chars)
            {
                yield return a.ToString();
                foreach (var b in GetCombinations(Chars, MaximumLength - 1)) yield return a + b;
            }
        }
    }
}
```

}With this code you can get absolutely all combinations of passwords, but it will take an incredible amount of time. I'm not saying that this is the best way (rather the opposite), but this is the real Brute Force.

Spoiler [\(Click to View\)](#)

```
Code using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
```

```
namespace Cycle
{
    class Program
    {
```

```

static void Main(string[] args)
{
    Console.Title = "Cycle";
    int Attempt = 0;
    Console.WriteLine("Enter the minimum password length:");
    int MinQuantity = int.Parse(Console.ReadLine());
    Console.WriteLine("Enter the maximum password length:");
    int MaxQuantity = int.Parse(Console.ReadLine());
    string Chars =
"aAbBcCdDeEfFgGhHilJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ0123456789";
    for (int i = MinQuantity; i <= MaxQuantity; i++)
    {
        foreach (var Password in GetCombinations(Chars.ToArray(), i))
        {
            Attempt++;
            Console.WriteLine("Attempt number:" + Attempt + " | Password:" + Password);
            StreamWriter file = new StreamWriter("Passwords.txt", true);
            file.WriteLine(Password);
            file.Close();
        }
    }
    Console.ReadLine();
}
private static IEnumerable<string> GetCombinations(char[] Chars, int MaximumLength)
{
    if (MaximumLength < 1) yield break;
    foreach (var a in Chars)
    {
        yield return a.ToString();
        foreach (var b in GetCombinations(Chars, MaximumLength - 1)) yield return a + b;
    }
}
}
}

```

Author: Mathian Developer

HFDone - December 26, 2018, 12:27 pm

better use Parallel.ForEach

.TheMarv - December 26, 2018, 7:48 pm

By outputting every password to console, you're massively reducing the speed of the program

Mathian - December 26, 2018, 9:03 pm

(Dec 26, 2018 08:48 PM).TheMarv Wrote: [\[CITE\]](#) By outputting every password to console, you're massively reducing the speed of the program Oh my God! Friend, you can not output to the console. I provided the generation algorithm

.TheMarv - December 27, 2018, 12:31 am

Code foreach (var Password in GetCombinations(Chars.ToArray(), i))

Console.WriteLine(Password);

This will output every possibility to the console, that's what I was referring to

Paskowsky - December 27, 2018, 2:36 am

@OP thank you for sharing this!

(Dec 27, 2018 01:31 AM).TheMarv Wrote: [\[CITE\]](#) Code foreach (var Password in
GetCombinations(Chars.ToArray(), i))
 Console.WriteLine(Password);

This will output every possibility to the console, that's what I was referring to
He has provided the algorithm to generate passwords, where you want them outputted is your
choice ?

Regards,
Paskowsky

.TheMarv - December 27, 2018, 2:40 am

(Dec 27, 2018 03:36 AM)Paskowsky Wrote: [\[CITE\]](#) @OP thank you for sharing this!

He has provided the algorithm to generate passwords, where you want them outputted is your
choice ?

Regards,
Paskowsky

Ofc, I just wanted to point out, that putting it out in the console would decrease the performance.

Mathian - December 27, 2018, 9:00 am

(Dec 27, 2018 01:31 AM).TheMarv Wrote: [\[CITE\]](#) Code foreach (var Password in
GetCombinations(Chars.ToArray(), i))
 Console.WriteLine(Password);

This will output every possibility to the console, that's what I was referring toYou can save to file,
add to database or apply. Conclusion in the console - a demonstration of work

(Dec 27, 2018 03:36 AM)Paskowsky Wrote: [\[CITE\]](#) @OP thank you for sharing this!

He has provided the algorithm to generate passwords, where you want them outputted is your
choice ?

Regards,
PaskowskyEnjoy :)

0xDEAD10CC - December 27, 2018, 2:20 pm

Enumerators and IEnumerable = very slow. An iterative approach would be much faster. This code
snippet is really not useful in the real world for relatively good bruteforcing.

Mathian - December 28, 2018, 2:59 am

(Dec 27, 2018 03:20 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Enumerators and IEnumerable = very slow.
An iterative approach would be much faster. This code snippet is really not useful in the real world
for relatively good bruteforcing.I by no means suggest to produce brutus by this method