C# Programming Exercise 15
Password Cracker

C# Step by Step

1 Introduction

In this exercise you will write a brute force password cracker. A "brute force" approach requires the mindless generation of all possible combinations and comparison of each combination with the target. In this exercise, you will have a user provided password, and you will match that password with the combinations you generate. You will then parallelize your cracker program to speed it up.

Allowable password characters consist of all printable characters on the keyboard. Printable characters include alphabetical characters, both upper case and lower case, digits, punctuation characters, and the other printable characters, e.g., !, @, #, $, %, ; &, *, (,), -, +, etc.

2 Part one, single threaded program — 80 points

Write a program that generates all possible combinations of printable characters. Prompt the user to enter a password, then match the password with your code. Start with a one character password. Wrap the brute force logic in a stopwatch object so you can time how long the program runs to match the password. Then, crack a two character password. Then add more characters, one at a time, until you hit the limit of a reasonable run time. On my computer, I can match a four character password in several minutes, but a five character password can take a 30 or 40 miutes.

3 Part two, multi-threaded program — 100 points

After Part one works, add parallelization so that your program executes in several threads. Add more characters to the password. Track the run time of your program, and see how many more characters you can add to the password cracker.

This is a non-trivial program. You will have several difficult problems to solve in order to geet this to work. I spent about four hours writing this prgram, and you may spend two or three times as long. However, this program is within your skill set, and you will learn a good deal about the actual implementation of a multithreaded application.