Decisions in Visual Basic II

Topics

- if/else statements
- testing the "randomness" of random numbers
- introduction to concept of string patterns
- creating your own test cases to verify the correctness of a program
- choosing appropriate numerical data types
- simple vs compound conditions

Password crack calculator

Disclaimer. The program that you develop for this exercise should **NOT** be considered a measure of the strength of a particular password. An obvious example would be the password '*P@ssw0rd*', which, according to the program, would take 147.53338 centuries to crack, but in practice would be among the first passwords checked by any cracker worth their salt.

3. Use **File Explorer** to navigate to the folder containing the final version of your program `PasswordCrackCalculator_Part2` in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as `PasswordCrackCalculator_Part3`. Launch VS Express and use it to open the program `PasswordCrackCalculator_Part3`.

   Now, let's address test8 and test9.

> Your program should behave identically to the solution which can be run by double-clicking the file `PasswordCrackCalculator_Part3.exe` found in the `Executables` folder inside of your lab folder.
>
> When done, save and close program `PasswordCrackCalculator_Part3`.

> Checkpoint 5 (80/100): A successful `PasswordCrackCalulator_Part3` program:
>
> - produces correct results for all test cases except test9
> - must also have successfully completed Checkpoint 2

4. Use **File Explorer** to navigate to the folder containing the final version of your program PasswordCrackCalculator_Part3 in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as PasswordCrackCalculator_Part4. Launch VS Express and use it to open the program PasswordCrackCalculator_Part4.

   Enhance your program so that along with all previous information it now also calculates and outputs the total number of seconds required to check the entire search space. The number of checks per second that the computer is able to do is a value that is input by the user. Be sure to include the units (seconds) in your output statement.

Your program should behave identically to the solution which can be run by double-clicking the file PasswordCrackCalculator_Part4.exe found in the Executables folder inside of your lab folder.

When done, save and close program PasswordCrackCalculator_Part4.

Checkpoint 6 (85/100): A successful PasswordCrackCalulator_Part4 program:

- output the correct amount of time to check the search space
- must also have successfully completed Checkpoint 2

5. Use **File Explorer** to navigate to the folder containing the final version of your program PasswordCrackCalculator_Part4 in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as PasswordCrackCalculator_Part5. Launch VS Express and use it to open the program PasswordCrackCalculator_Part5.

   To make your program more user friendly, it should present the time to search using the most appropriate unit of measurement from the following list:

   - seconds,
   - minutes,
   - hours,
   - days,
   - weeks,
   - years, or
   - centuries.

   For example, if the number of seconds to search the space is less than 60, it should use the unit *seconds*, however if it is greater than 60, but less than 60×60=3600, then it should use the unit *minutes* and so on, all the way up to *centuries*. To keep things consistent, we will define a year as 52 weeks.

> Your program should behave identically to the solution which can be run by double-clicking the file PasswordCrackCalculator_Part5.exe found in the Executables folder inside of your lab folder.
>
> When done, save and close program PasswordCrackCalculator_Part5.

> Checkpoint 7 (90/100): A successful PasswordCrackCalulator_Part5 program:
>
> - correctly converts and outputs time units
> - must also have successfully completed Checkpoint 6

6. Use **File Explorer** to navigate to the folder containing the final version of your program PasswordCrackCalculator_Part5 in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as PasswordCrackCalculator_Part6. Launch VS Express and use it to open the program PasswordCrackCalculator_Part6.

Your program should behave identically to the solution which can be run by double-clicking the file PasswordCrackCalculator_Part6.exe found in the Executables folder inside of your lab folder.

When done, save and close program PasswordCrackCalculator_Part6.

Checkpoint 8 (100/100): A successful PasswordCrackCalulator_Part6 program:

- handles invalid input (passwords that contain characters not allowed), by output a meaningful error message
- must also have successfully completed Checkpoint 7

Submission Instructions

Your M:\CS130\Labs\Lab04_YourLastName_YourFirstName folder should contain your solutions to this and Tuesday's lab.

To submit your work, copy this folder and paste it to N:/Handins/CS130/Lab04 PRIOR to the end of lab today. Late submissions won't be accepted and will receive a grade of 0 instead.