**80%**

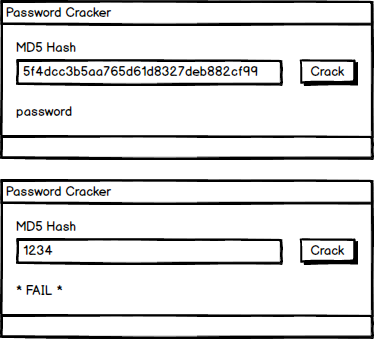
**All code submitted must be your original work. Code must be uploaded to GitHub in the HandsOnTests/Ch12/ folder**

**EX1 [50 pts]**

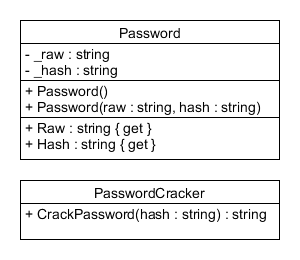
A common way to crack MD5 hashed passwords is to use a rainbow table. Below is a short rainbow table for some common passwords. Use this table to build a basic password cracker.

You can quickly generate MD5 hashes yourself using this website:

http://www.miraclesalad.com/webtools/md5.php



|  |  |
| --- | --- |
| **Password** | **MD5 Hash** |
| 123456 | e10adc3949ba59abbe56e057f20f883e |
| 123456789 | 25f9e794323b453885f5181f1b624d0b |
| qwerty | d8578edf8458ce06fbc5bb76a58c5ca4 |
| 111111 | 96e79218965eb72c92a549dd5a330112 |
| password | 5f4dcc3b5aa765d61d8327deb882cf99 |
| qwertyuiop | 6eea9b7ef19179a06954edd0f6c05ceb |
| 123321 | c8837b23ff8aaa8a2dde915473ce0991 |
| google | c822c1b63853ed273b89687ac505f9fa |
| P@ssw0rd | 161ebd7d45089b3446ee4e0d86dbcf92 |
| Tr0ub4dor&3 | 4ece57a61323b52ccffdbef021956754 |

**Features**

1. The user can enter an MD5 hash.
2. If the hash is in the table then display the raw password.
3. If the hash is not in the table then display "\* FAIL \*"

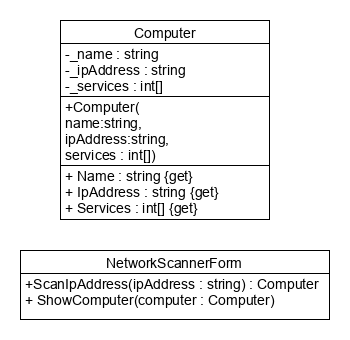
**Form Methods**

**CrackPassword()** A method defined **in the form** that takes an MD5 hash and returns the raw password, if the hash is in the table. If the hash is not in the table then it should return null.

**EX2 [50 pts]**

A network scanner is a common tool for finding security gaps in a network. Create a prototype that accepts an IP address and displays information about the machine.

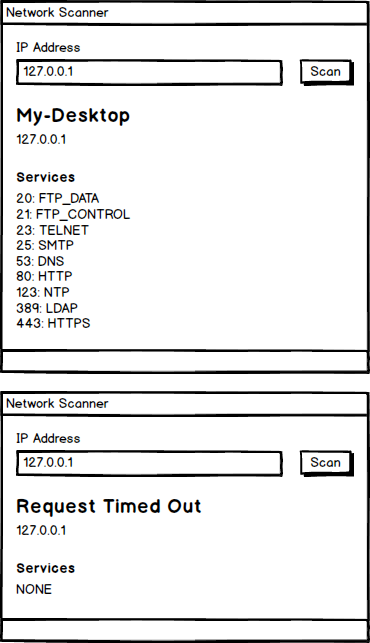
|  |  |  |
| --- | --- | --- |
| **Computer Name** | **IP Address** | **Services** |
| My-Desktop | 127.0.0.1 | 20, 21, 23, 25, 53, 80, 123, 389, 443 |
| google-public-dns-a | 8.8.8.8 | 53 |
| ranken.edu | 47.44.246.80 | 25, 80, 443 |



**\*Note\* - the Services field & Property of the Computer class can optionally be made a string**

**Features**

1. The user can enter an IP address.
2. If the IP address is in the table then display the available computer info.
3. If the IP address is not in the table then display the output as shown.

**Form Methods**

**ScanIpAddress()** A method defined **in the form that** takes an ip address and returns the computer with that ip address. Or null if no computer has that ip address.

**ShowComputer()** A method defined **in the form that** takes the results of the scan and displays the appropriate info.

**Rubric EX1**

* Controls are laid out as expected – 5 pts
* Tab order is configured – 5 pts
* Control names follow naming conventions – 5 pts
* Variable names follow naming conventions – 5 pts
* UI implements all requested functionality - 15 pts
* **Password** class developed and implemented to solve the problem per UML specifications – Class/objects not implemented in solution
  + Fields - 3pts
  + Constructors - 4pts
  + Properties - 3pts
* **CrackPassword()** method implemented as described -5pts

**Rubric EX2**

* Controls are laid out as expected – 4 pts
* Tab order is configured – 4 pts
* Control names follow naming conventions – 3 pts
* Variable names follow naming conventions – 4 pts
* **Computer** class developed and implemented to solve the problem per UML specifications
  + Fields - 3pts
  + Constructors - 4pts
  + Properties - 3pts
* **ScanIpAddress()** method implemented as described - 5 pts – should return a computer object
* **ShowComputer()** implemented as described - 5 pts – should accept a computer object
* UI implements all requested functionality - 15 pts