Working Makefile included

Program builds without errors or warnings

Program is readable, modular, well documented, and reasonably efficient, as well as correct:

* Readable
  + Descriptive variable names
  + Descriptive function names
  + Limited (no) use of obfuscated code
  + Appropriate internal comments
* Modular
  + Functions perform a single task
  + Classes are general and reusable
  + Functions interact only through interfaces
* Well documented
  + Doxygen main page describing overall program
  + Classes described -2
  + Class member functions documented -2
  + Doxygen runs without errors or warnings
  + Doxygen generates navigable documentation
* Reasonably Efficient
  + Program doesn’t leak memory

Cracker is functional:

* Level 0 40/40
* Level 1 10/10
* Level 2 10/10
* Level 3 10/10
* Level 4 (a, b, or c) 15/15
* Level 5 (a, b, or c) 15/15
* Level 6 6/5
* Level 7 0/5
* Working, valid vault 5/10

Functionality Score : 107/100

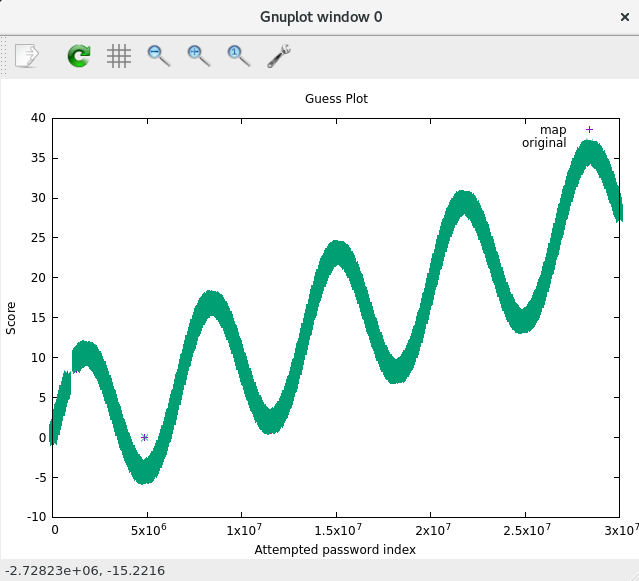
Documentation: No deductions

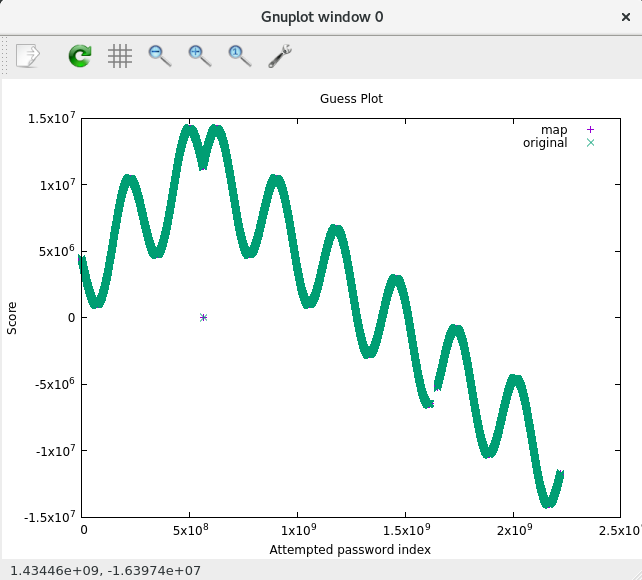
This project did not require a tremendous amount of code but it did require a fair amound of thought, design, and innovation.

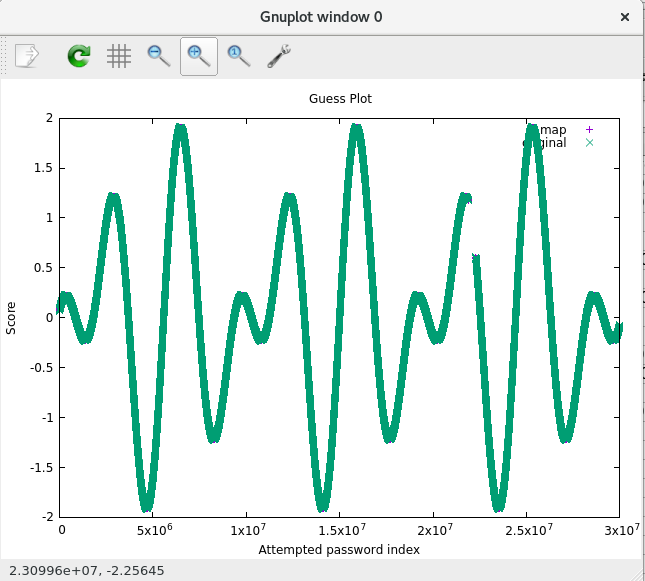
A good submission, gentlemen. Did you notice how your documentation was scrambled due to having two classes named Cracker in two different files (cracker.cpp and cracker2.cpp)? Seems like some mention of that should have been made in the ToDo or Bugs section. The inheritence diagram is also incorrectly generated due to this duplication.

One thing that concerns me is the repository activity which seems to indicate that one team member did the majority of the work. I know that some paired and team programming was done but the repository activity shows way more lop-sidedness then I would expect. Part of team programming should involve a clear division of labor and solo work when appropriate. Having two or three people sitting around a single keyboard / monitor is good for debugging sessions but not for the entirety of the project.

Even though your vault was quite simple it does appear to be working and valid so I gave you some points for spending the time to make it work. I also had some custom vaults that I had created and other teams have submitted with which I used to test your vault. Unfortunately, your cracker was not successful in cracking those unknown vaults which means it may have ‘learned’ the development vaults but in the process lost its ability to deal with a function that was similar but not identical (see plots below).







|  |  |
| --- | --- |
| Level0 | 1.702186 |
| Level1 | 0.02211195 |
| Level2 | 0.02250741 |
| Level3 | 0.1413575 |
| Level4a | 0.03069076 |
| Level4b | 0.0433716 |
| Level4c | 0.04303375 |
| Level5a | 1.4835993 |
| Level5b | 1.5676767 |
| Level5c | 0.0892136 |
| Level6 | 1.997435 |
| Level7 | 9 (just means this timed out 3 seconds each) |