University of West Florida
Department of Computer Science
Data Structures and Algorithms II
Project #1

The Project

The goal of this project is to create an efficient storage and lookup scheme for userids and passwords utilizing an externally chained hash table. Programming will be object-oriented using C++. The assignment has two parts.

Part 1, Creating encrypted passwords.

We will use the names in "`names.txt,`" located with this project description, as userids. You will only need the data in the first column of the file. You will first generate random passwords for each userid and write them with the corresponding userid to a raw file, one userid, password combination per line. This file should be named "`rawdata.txt.`" Note that we DO NOT want sequential passwords such as a1, a2, a3, a4, etc. Passwords should be 9 characters, comprised of random combinations of lower-case letters (admittedly crummy, but simpler for our purposes). Userids are unique, passwords do NOT need to be unique.

Next, you will read the unencrypted file, encrypt the passwords, and write this file back to a disk file named "`encrypteddata.txt.`" We will use the Vigenere Cipher to create encrypted passwords:

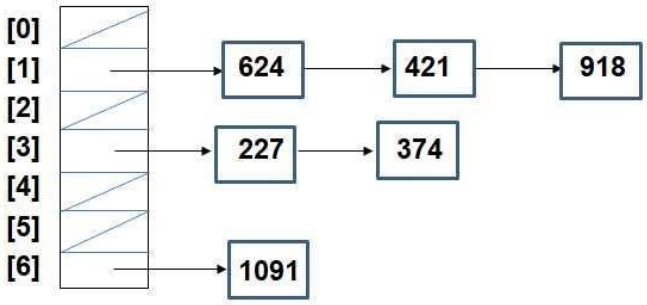
Use ***jones*** as the key:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
| | | | | | | | | | | | | | | | | | | | | | | | |
j k l m n o p q r s t u v w x y z a b c d e f g h i

a b c d e f g h i j k l m n o p q r s t u v w x y z
| | | | | | | | | | | | | | | | | | | | | | | | |
o p q r s t u v w x y z a b c d e f g h i j k l m n

a b c d e f g h i j k l m n o p q r s t u v w x y z
| | | | | | | | | | | | | | | | | | | | | | | | |
n o p q r s t u v w x y z a b c d e f g h i j k l m

a b c d e f g h i j k l m n o p q r s t u v w x y z
| | | | | | | | | | | | | | | | | | | | | | | | |
e f g h i j k l m n o p q r s t u v w x y z a b c d

a b c d e f g h i j k l m n o p q r s t u v w x y z
| | | | | | | | | | | | | | | | | | | | | | | | |
s t u v w x y z a b c d e f g h i j k l m n o p q r
```

You would then encode **data** : **moge**
The Vigenere Cipher is a simple cypher that solves the problem of using frequency analysis to defeat a basic encryption scheme. In this method, you first choose a key that you use to make a mapping of real letter to cypher by adjusting the alphabet according to the key:

Part 2. Testing userid/password combinations

Now we get to the hashing part. You will build an externally chained hash table of userids and passwords. You will process "`encrypteddata.txt`" to create the hash table by hashing on userid. At that appropriate location, you will store the unencrypted userid and encrypted password. Recall that an externally chained hash table is pretty easy to implement as simply a collection of linked lists:



Your program will then automatically test:

- 5 legal userids and passwords by testing 5 randomly selected entries in "`rawdata.dat.`" The idea is to hash into the table by userid, find the userid in the table, encrypt the plaintext password, and show that it matches the password on that entry.
- 5 legal userids and ILLEGAL passwords by testing the same 5 entries from the previous part in "`rawdata.dat.`" The idea is to hash into the table by userid, find the userid in the table, change one letter in the unencrypted password, encrypt the now illegal plaintext password, and show that it does not match the password on that entry.

No user interaction should be required by the program.

Output:
Write the five legal and illegal userid and password combinations you tried to the console including an indication of the match:

```
Legal:
Userid      Password   Result
```

```
SMITH       asdfvfrty  match
JOHNSON     okmnjuygt  match
WILLIAMS    yhnbgtrde  match
JONES       wsxzaqwer  match  BROWN
bhvfgtyui  match

Illegal:
Userid      Password    Result
SMITH       xsdfvfrty  no match
JOHNSON     xkmnjuygt  no match
WILLIAMS    xhnbgtrde  no match
JONES       xsxzaqwer  no match
BROWN       xhvfgtyui  no match
```

Note: I am aware of `JournalDev, thecrazyprogrammer.com, tutorialsPoint` and other solutions for Vigniere Cypher in C++. DO NOT USE THEM. WRITE YOUR OWN SOLUTION!

About the use of GitHub:

I STRONGLY suggest that you create GitHub repos for the projects in this course and that you routinely pull the current version of your project and push updates. You should do this because:
1. You will use configuration management tools of some sort from the first day you are working in the world.
2. If any copying of assignments is detected, we can trace your record of commits to decide who copied from whom.

I STRONGLY suggest that you employ a test-driven development approach because:
1. You are likely to encounter a development culture in which it is encouraged or required.
2. It will force you to do incremental development and always have a running program.

I will not produce test cases to which you might code because
1. that will be your responsibility in the real world
2. test cases imply a detailed design of the project. At this level, you must be able to produce your own designs.

Deliverables

1) Makefile
2) User's manual

3) UML diagram
4) C++ Source Code (.hpp and .cpp)
5) `names.txt`

Submission

Your project must be submitted using the instructions below to be graded.

1. Compile and run your program one last time before submitting it. Your program must run in Linux in the multiplatform lab.
2. Place all deliverables in a single folder that is named with your last name and first name initial. For example, if your name is John Coffey, the folder should be coffeyj. If you are a MacOS user, delete the _MacOS_ folder before zipping up your submission.
3. Create a "zip" file (using WinZip or similar program) to hold your project files. The name of your zip file should be named exactly like the single folder inside it but with the .zip extension. For instance: coffeyj.zip.
3. Login to Canvas and select our course from the dashboard.
4. Go to the Assignments page and choose the assignment for which you are submitting a project.
5. Click the "Submit Assignment" button.
6. Use the "Choose File" capability to select and upload your .zip file.

Please review the policy on ACADEMIC MISCONDUCT. This is an individual assignment.