Assignment 1 (CS458)

Due: 11:59pm Sept. 25 (Sunday)

This assignment is done individually. You can use C/C++/Java/Python to implement this assignment.

Goal :

1.Learn how to write and use makefile (C/C++/Java):
http://www.delorie.com/djgpp/doc/ug/larger/makefiles.html (C)
http://www.cs.swarthmore.edu/~newhall/unixhelp/javamakefiles.html (Java)

2. Implement the autokey cipher studied in the class.

Description:

In this assignment, you will implement the autokey cipher studied in the class, to encrypt and decrypt files containing only lower-case letters a-z. **The file does not contain spaces, tabs, and newlines.** You can assume that the file contains less than 1000 characters. The key is "security", which will be hardcoded in your program. Your executable will take the following arguments.

- inputfile: input file name
- outputfile: output file name
- 1/0: encryption/decryption

Please note that the file name does not need to be end with .txt (e.g., in, in.txt, in1, etc. are valid file names).

If you use C, your Makefile **must** give the executable code the name **auto**. If you use Java, your Makefile will generate a file Auto.class. If you use python, your python file should have name auto.py.

For example, if you use C, your code will execute in the following form:

```
./auto <inputfile> <outputfile> 1/0
```

- auto in out 1: encrypt file "in" using the autokey cipher with key "security" and store the result in file "out".
- auto out in1 0: decrypt file "out" using the autokey cipher with key "security" and store the result in file "in1".

After executing the above commands, the file content in files "in" and "in1" should be the same.

Submission guideline

- You need to hand in your **source code** and **Makefile (C/C++/Java)** electronically (please **do not submit .o or executable code**). Your code should compile and run correctly on remote.cs.binghamton.edu.
- Submit one **README** file (text file, do not submit a .doc file) which contains
  - You name and email address.
  - Whether your code was tested on remote.cs.binghamton.edu.

  - How to execute your program.
  - (Optional) Briefly describe your algorithm or anything special about your submission that the TA should take note of.

- Place the source code, the readme and Makefile under one directory with a unique name (such as p1-[userid] for assignment 1, e.g. p1-pyang).

- Tar the contents of the above directory using the following command.
  tar –cvf [directory_name].tar [directory_name]
  E.g. tar -cvf p1-pyang.tar p1-pyang/

Submit the tared file created through brightspace.binghamton.edu.

Grading guideline

- Correctness (C/C++/Java): 88'
- Correctness (Python): 96'
- Readme, correct format of execution: 4'
- Makefile (C): 8'

Academic Honesty

All students should follow Student Academic Honesty Code (**if you have not already read it, please read it carefully**). All forms of cheating will be treated with utmost seriousness. You may discuss the problems with other students, however, you must write your OWN codes and solutions. Discussing algorithms and code is NOT acceptable. Copying an assignment from another student or allowing another student to copy your work may lead to the following:

- **Report to the department and school**
- **0 in the assignment or F in this course.**

Moss will be used to detect plagiarism in programming assignments. You need ensure that your code and documentation are protected and not accessible to other students. Use **chmod 700** command to change the permissions of your working directories before you start working on the assignments. If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate.

Students should also read the Watson Faculty Letter to students. Here is the link to it: https://docs.google.com/document/d/1J-9taFB4Vz_vu9tyPwphFFzGwmV6yos5/edit?usp=sharing&ouid=101123343507519900554&rtpof=true&sd=true