

Assignment 1 (CS558)

Due: 11:59pm Sept. 24 (Sunday)

This assignment should be done individually.

You can use **C, C++, Java, or Python** in this assignment.

Goal:

1. Learn how to write and use makefile.

<http://www.delorie.com/djgpp/doc/ug/larger/makefiles.html> (C/C++)

<http://www.cs.swarthmore.edu/~newhall/unixhelp/javamakefiles.html> (Java)

2. Implement the [row transposition cipher](#).

Assignment Description:

In this assignment, you will implement the row transposition cipher for encrypting and decrypting files containing lower-case letters (a-z) and digits (0-9). The file contains a single line without spaces and tabs. You can assume that the file contains fewer than 1000 letters and digits.

To simplify the encryption process, we will use “z” as paddings when needed. For instance, when a single filler is needed, one “z” will be used as the padding. If two fillers are needed, then two “z”s will be used.

You do NOT need to remove fillings when decrypting a ciphertext.

Your program should have five arguments with the following order:

<keylength> <key> <inputfile> <outputfile> <enc/dec>

- **keylength**: the length of the key (e.g., 6)
- **key**: the key used for encryption and decryption (e.g., 315462)
- **inputfile**: input file name (e.g., input)
- **outputfile**: output file name (e.g., output)
- **enc/dec**: encryption/decryption (i.e., enc means encryption and dec means decryption)

Error handling: If any of the following conditions are not satisfied, your program will display an error message and terminate.

1. <keylength> must match the length of <key>.
2. <key> must include all digits from 1 to <keylength> with each digit occurring exactly once.
3. <inputfile> must contain only lowercase letters (a-z) or digits (0-9).

If you are using C, your Makefile should produce an executable named **trans**. If you are using Java, your Makefile should compile your code into a file named **Trans.class**. If you are using python, your python script should be named **trans.py**.

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

```
./trans 7 2436175 in out enc:
    encrypt file in with key 2436175 and store the result in file out; the length of the key is 7.
./trans 7 2436175 out in1 dec:
    decrypt file out with key 2436175 and store the result in file in1; the length of the key is 7.
```

After executing the above two commands, files `in` and `in1` should contain the same content except that file `in1` may contain fillers “z” at the end of the line.

Compiling your program:

If you are using C/C++/Java, please write a Makefile to compile your program. When executing the “make” command, your program should be compiled.

Submission guideline:

Please hand in your **source code**, a **README**, and a **makefile (C/C++/Java)** electronically via brightspace.binghamton.edu (**please do not submit executable code**). Your code should compile and run correctly on remote.cs.binghamton.edu.

- The **README** file (**text file, do not submit a .doc file**) should contain
 - Your name and the email address
 - The programming language used
 - Whether your code was tested on remote.cs.binghamton.edu.
 - How to compile and execute your program.
 - (Optional) Anything special about your submission that you would like the grader to take note of.
- Place all your files under one directory with a unique name (such as `p1-[userid]` for assignment 1, e.g. `p1-pyang`).
- Tar the contents of this directory using the following command.
tar -cvf [directory_name].tar [directory_name]
E.g. `tar -cvf p1-pyang.tar p1-pyang/`
- Use brightspace.binghamton.edu to upload the tared file you created above.

Grading guideline

- Makefile (C/C++/Java): 6'
- README and the correct format of execution (i.e., the executable name, and the number and order of arguments should conform to the assignment description): 6'
- File I/O: 15'
- Correctness of encryption and decryption (C/C++/Java): 58'
- Correctness of encryption and decryption (Python): 64'
- Error handling: 15'

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.

The use of ChatGPT or other AI tools is prohibited in this assignment.

If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate.