Task 2 – Linux practice – Can be done in pairs (NOT triples)

The task is made of 3 parts: files operations, dynamic libraries, and basic shell

Part A (24 pts):

You are requested to implement **two** small programs, that fills like a regular CMD tools.
Tool 1: "**cmp**"
the tool will compare two files, and return "0" if they are equal, and "1" if not (return an INT).
The tool will support -v flag for verbose output. By this we mean that the tool will print "equal" or "distinct", in addition to returning the int value.
The toll will support -i flag, that mean "ignore case" so "AAA" and "aaa" meaning equals

usage: cmp <file1> <file2> -v
output: equal

Tool 2: "**copy**", the tool will copy a file to another place and/or name.
The tool will return "0" on success, or "1" on failure (return an INT)
The tool will create a new file, if it does not exist, but it will not overwrite a file if it do exist.
the tool will support -v flag, that will output "success" if the file is copied, or "target file exist" if this is the case, or "general failure" on some other problem (in addition to the returned INT value).
The tool will support -f flag (that means force), that allows to overwrite the target file

usage copy <file1> <file2> -v
output: success

note: if the must-have parameters are missing (the file names) return 1, and print a usage explanation and the optional flags

Part B (24 pts):
You are requested to implement a coding library. We have two codding methods.
Method a, named **codecA**: covert all lower case chars to upper case, and all upper case to lower case. All other chars will remain unchanged.
Method b, named **codecB**: convert all chars to the 3-rd next char (adding a number of 3 to the ascii val).

The libraries should support "encode" and "decode" methods.
Note: the libraries should be "reversable", meaning that if one does "encode" and then "decode, he will get the original string

1) Write 2 different shared libraries , each implementing it's algorithm
2) Write a two tools, named **encode** and **decode**, to utilize the libraries. The tools will get some text and convert it according to selected library.
Usage : encode/decode <codec> <message>
output: encoded/decoded string

example: "encode codecA aaaBBB" will return "AAAbbb"
example: "decode codecB EEEddd" will return "BBBaaa"

Part C (52 pts):
You are requested to write a shell program **named stshell** (st for students).
The features are:

1) Be able to run CMD tools that exist on system (by fork + exec + wait)
2) Be able to stop running tool by pressing Ctrl+c, but not killing the shell itself (by signal handler)
3) Be able to redirect output with “>” and “>>”, and allow piping with “|”, at least for 2 following pipes. For example command like this should be supported “ls -l | grep aaa | wc”
4) be able to stop itself by “exit” command

Administrative details:
Doc Version: 1.1 updated on 02/04/23
Publication date: 02/04/23
Submission date: 23/04/23 8AM
What you need to submit:
1) all the **code C** files that have been used (I count at least 7)
2) a **Make** file for your solution, including **default** and **clean** options,that will generate execution files, **named as specified in the task**
3) a plaint text file (aka read.me) that will explain the usage of each tool and
your system environment (flavot of Linux).
Also a screenshot may be added (but not obligatory)

The task intends to be checked automatically. Therefore:
1) Please, don’t add any unnecessary folders, or files, as the submission may be ignored
2) You should provide a **ZIP** archive (nor rar/7z/tar etc.). Files in a wrong archive will not be checked
3) A **working make** file is a MUST for the task. Broken make (for any reason), will cause the whole work to get a zero mark, even if the code is perfect.
Beware of using mac OS, windows subsystem, and so on, as they may append changes
4) The executable should be named according to the task. Wrong name may lead to a zero (0) mark.
5) Don’t wait for the last day to submit your work. No excuses like “pc fall from 5-th flor, water ruined the memory, windows update brooked the file system” and so on will be accepted.
Verify on the moodle website, that the work is submitted. You may (but not have to) provide a screenshot of the submission.
6) If you are late with the submitting and have a good reason (Reserves, seeks, wedding, etc.), you have to provide evidence for that, and you can not submit with another person who is your pair, but don’t have any reason being late. The penalty for being late without a reason will be 5 points for a day, and you MUST update your trainer about that, and follow the instruction if they exist.
7) A screenshot from camera/phone is not accepted