done seriously.

Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.

- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that "git clone" is used in an empty folder.

- Check carefully that no malicious aliases were used to fool you and make you evaluate something that is not the content of the official repository.

- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).

- If you have not completed the assignment you are going to evaluate, you have to read the entire subject before starting the evaluation process.

- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth.
In these cases, the evaluation process ends and the final grade is 0, or -42 in case of cheating. However, except for cheating, student are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

Attachments

Mandatory part

The command ./pipex file1 cmd1 cmd2 file2 must behave like this command : < file1 cmd1 | cmd2 > file2

Error and arguments management

- The program takes 4 arguments, no more, no less
(except for bonus part) and only in the right order.
- Error management is correct: existing files, files rights,
the binary of the command exists etc.

If these points are respected, check `Yes` and continue the evaluation.
Otherwise, the evaluation is overuse Incomplete work or the appropriate flag.

Yes | No

The program

The program does what is requested, without displaying any additional
information/steps against the shell command

Run your own tests et compare the results of shell exit and
of shell output and that of the program.
If you haven't any idea, look at the subject examples.

If no error is detected, check `Yes` and continue.

Yes | No

Bonus

Evaluate the bonus part if, and only if, the mandatory part has been entirely and perfectly done, and the error management handles unexpected or bad usage. In case all the mandatory points were not passed during the defense, bonus points must be ignored.

Bonus

The program can copy the use of << and >>.
test multiple times something like :
CMD << STOP_VALUE | CMD1 >> file1

Yes | No