File Processing Application

**Task description:**

Develop a command-line application in Perl that allows the user to manipulate text files with number content. The final solution should provide the following:

1. Handling of *.txt* files that contain lines with numbers. Each number is presented as a sequence of digits (0-9) and must not start with 0. The numbers in each line are separated by spaces or tabs. Each line must start with a number, but not space or tab.

*Example of the file content:*

*112312321525 12312312 5234234234 231321321312 123123 2131353453435345345*

*213123523520234234 2342340005320 3240230042340230803240 3240909*

*9897823423975239*

2. Command-line application that asks the user for a file (*txt* format).

3. After the user specified the path to the file the application should provide the following options:

a. Validate the file contents – check if the file contents are valid, based on the restrictions in 1. and present to the user the result of the validation (the format is up to you, for instance: *“line <number>, character <number> is not allowed”* or *“number at line <number> starts with 0”*).

b. Switch entire line from the file with an entire other line. The user should specify two line indexes and the application should switch these lines.

*Example user input: <first\_line\_index> <second\_line\_index>*

c. Switch number at specific index in one line with a number with specific index from another line. The user should provide four indexes and the application should switch the number from the first line with the number from the second line.

*Example user input:*  *<first\_line\_index> <first\_line\_number\_index> <second\_line\_index><second\_line\_number\_index>*

d. (Related to both b. and c.) The result file (with the switches) should be saved in the same file that is initially specified by the user. Initial validation should also be performed as described in a. and if the file is with invalid content any switch operation should not be performed and the validation status should be printed. Proper validation of the user-provided indexes should also be ensured.

(Optional functionality)

e. Apply ‘CRUD’ operations on a selected position of a number:

- Insert a number at a given position in the file. The user should provide the two indexes at which the number should be inserted and the value of that number. The number should be inserted in that position and the content of the row from that position to it’s end should be shifted with one index to the right. *Example user input:*  *<line\_index> <line\_number\_index> <number to be inserted>*

- Read a number at a given position. The user should provide the two indexes of a number to be red. The application should print that number to the user. *Example user input:*  *<line\_index> <line\_number\_index>*

- Modify a number at a given position. The user should provide the two indexes of the number to be modified and a value. The value given by the user should replace the existing value at that position. *Example user input:*  *<line\_index> <line\_number\_index> <number to be set>*

- Remove a number at a given position. The user should provide the two indexes of the number to be removed. All remaining numbers in that row should be moved with one index to the left. *Example user input:*  *<line\_index> <line\_number\_index>*