1-CodeManipulator:

a-Inputs:

- code: string that carry user's code to apply the operation on it. - steps: string carry number of steps of operation.

- Inc: Boolean indicates type of operation whether increment takes (true) value or decrement (false) value.

b-Outputs:

- string carry the resulted code from operation.

Functionality: takes the code and increment it or decrement it according to Inc variable by number of steps.

2-increment:

a-Inputs:

- code: string that carry user's code to apply the increment on it. - step: integer carry number of steps of increment.

b-Outputs:

- string carry the resulted code from the increment of step.

Functionality: takes the code and increment it the number of steps.

3-IncCharNum:

a-Inputs:

- code: string that carry user's code to apply the increment on it. - step: integer carry number of steps of increment.

- ind: index of the numerical char that should be incremented.

b-Outputs:

- string carry the resulted code from the increment of the numerical char by the needed value from step.

Functionality: increment the numerical char by max value the char can be added by. Remove the value added from step. Extend the length of code if the char was last one in code & step didn't reach zero.

4-IncCharAlpha:

a-Inputs:

- code: string that carry user's code to apply the increment on it. - step: integer carry number of steps of increment.

- ind: index of the Alphabetical char that should be incremented.

b-Outputs:

- string carry the resulted code from the increment of the Alphabetical char by the needed value from step.

Functionality: increment the Alphabetical char by max value the char can be added by. Remove the value added from step. Extend the length of code if the char was last one in code & step didn't reach zero.

5-decrement:

a-Inputs:

- code: string that carry user's code to apply the decrement on it. - step: integer carry number of steps of decrement.

b-Outputs:

- string carry the resulted code from the increment of step.

Functionality: takes the code and decrement it the number of steps.

6-DecCharNum:

a-Inputs:

- codearr: array that carry user's code to apply the decrement on it. - step: integer carry number of steps of decrement.

- alpha: array carry all alphabetical letters.

- ind: index of the numerical char that will be decremented.

- IsDecremented: Boolean that determines whether decrement process finished yet.

b-Outputs:

- array carry the resulted code from the decrement of the numerical char.

Functionality: takes the code and decrement the numerical char by the value it's position should be decremented. If this value larger than value of the char it barrow from next chars value to make the numerical char reach max & modify the code. Update step. Set whether process finished or not using (IsDec…)

7-DecCharAlpha:

a-Inputs:

- codearr: array that carry user's code to apply the decrement on it. - step: integer carry number of steps of decrement.

- alpha: array carry all alphabetical letters.

- ind: index of the alphabetical char that will be decremented.

- IsDecremented: Boolean that determines whether decrement process finished yet.

b-Outputs:

- array carry the resulted code from the decrement of the alphabetical char.

Functionality: takes the code and decrement the alphabetical char by the value it's position should be decremented. If this value larger than value of the char it barrow from next chars value to make the alphabetical char reach max & modify the code. Modify step. Set whether process finished or not using (IsDec…)

8-CharDecremented:

a-Inputs:

- codearr: array that carry user's code to apply the decrement on it. - step: integer carry number of steps of decrement.

- ind: index of the alphabetical char that has been decremented.

- IsDecremented: Boolean that determines whether decrement process finished yet.

b-Outputs:

- array carry the code but after making all values of it's chars zeros if the char of index is last char otherwise return same array inputted.

Functionality: do common repeated operations after any char value is decremented. Check whether the decrement process finished if so finish process using (IsDec…),otherwise check whether the char decremented was last char in code so the decrement value will be larger than code value so resulted code become zero and it finish process, otherwise refers the index ind for next char.

9-getnext:

a-Inputs:

- codearr: array that carry user's code to apply the decrement on it.

- alpha: array carry all alphabetical letters.

- ind: index of the numerical char that will be decremented.

- IsDecremented: Boolean that determines whether decrement process finished yet.

- Dec: integer value that should be decremented from the char that it's value less than this integer.

- NextOldValue: saves old value of the char before it get maximized from the getnext function.

b-Outputs:

- array carry the resulted code after maximizing the char by values from next chars value or zero value code if there's no next chars has value more than one.

Functionality: maximize the char value so it could be decrement by dec from values of next chars or if theirs no chars the code ceases.

10-updatecode:

a-Inputs:

- codearr: array that carry user's code to apply the decrement on it.

- alpha: array carry all alphabetical letters.

- ind: index of the numerical char that will be decremented.

- IsDecremented: Boolean that determines whether decrement process finished yet.

- Dec: integer value that should be decremented from the char that it's value less than this integer.

- NextOldValue: saves old value of the char before it get maximized

- getnextInd: index of the char that should get from it value. – MinusOne: integer value of the char that will be decrmented by one.

- IsChanged: Boolean determine whether the MinusOne decremented or not.

- IsGetted: Boolean determine whether the getting from next process is finished or not.

b-Outputs: array carry same code or code it's value is zero that theirs no chars to get values from them.

Functionality: decrease the MinusOne by one(value of the next char) & maximize all chars before that next char until the char that should be decremented & set that the value of MinusOne changed & that getting process is done. OtherWise refer index if there's one. else it make value of code zero & end whole decrmentation process.

11-max\_before:

a-Inputs:

-codearr: array carry the code that will be maximized. -ind: index of last char that will be maximized.

-getnextInd: index of the char before first char that will be maximized.

b-outputs:

- array carry code after maximized.

Functionality: array code values of its chars maximized from char after char of index getnextInd to char of index ind. 12-reset:

a-Inputs:

-code: string carry code that it's chars values will be zero.

-ind: index of the char that will start from it minimizing till end of string.

b-Outputs;

-string code minimized from index ind till end of code.

Functionality: make value of chars of code is zero starting from char of index ind till end of string.

13-CodeValidation:

a-Inputs:

- code: string code that will determine whether it's valid or invalid.

- ind: integer that will save index of first invalid char otherwise make it (-1).

-err\_id: integer saves the id of an error type if it occur or it will be 0.

-err\_message: string saves the error message if it occurs otherwise give "success".

b-outputs:

-boolean (true) if valid & saves ind(-1) , err\_id(0) & err\_message("success").

(false) if invalid & saves index of char that cause error, error id in err\_id & error message in err\_message.

Functionality: determine whether code valid or invalid & save message describe type of error if it occur with index of invalid char.

14-CodeRange:

a-Inputs:

- code: string code that will determine whether it's chars within validation range or not.

- ind: integer that will save index of first invalid char otherwise make it (-1).

-duplication: Boolean determine whether separator duplication occurs (true) otherwise gives (false).

b-outputs:

- boolean (true) if string chars all within validation range & saves ind(-1) & set duplication by (false).

-(false) if there's char not within validation range & saves index of char that cause error & set duplication by (true) if separator duplication caused error.

Functionality: determine whether code chars are within validation range or not , save index of first char that out of range & determine whether error caused by separator duplication.

Note: validation range: Chars : (alphabets, numbers, dashes, slashes & no separator duplication)

15-StepValidation:

a-Inputs:

- step: string code that will determine whether it's valid or invalid.

-err\_id: integer saves the id of an error type if it occur or it will be 0.

-err\_message: string saves the error message if it occurs otherwise save "success".

b-outputs:

-boolean (true) if valid & saves err\_id(0) & err\_message("success").

-(false) if invalid & saves error id in err\_id & error message in err\_message.

Functionality: determine whether step valid or invalid & save message describe type of error if it occurs.

16-StepRange:

a-Inputs:

- step: string step that will determine whether it's chars are numbers or not.

b-outputs:

- boolean (true) if string chars all are numbers.

-(false) if there's char that not a number.

Functionality: determine whether the step string consists of numbers only or not.

Limits: the limits of the inputs for CodeManipulator function & also using the validations methods prevent its occurrence.

1-code string: -it's length doesn't exceed 50 & not less than or equal zero.

-consists of slashes, dashes, alphabets & numbers only.

-doesn't start or end with separator.

-doesn't contain separator duplication.

2-step string: -consists of numbers only.

-it's numeric value less than 50 & not equal or smaller than zero.

sed errorrue) if on range d r that cause error, error id in err\_id & error mes