**NAME: SODIQ HAMIDU OMEIZA**

**MATRIC NO: 222514**

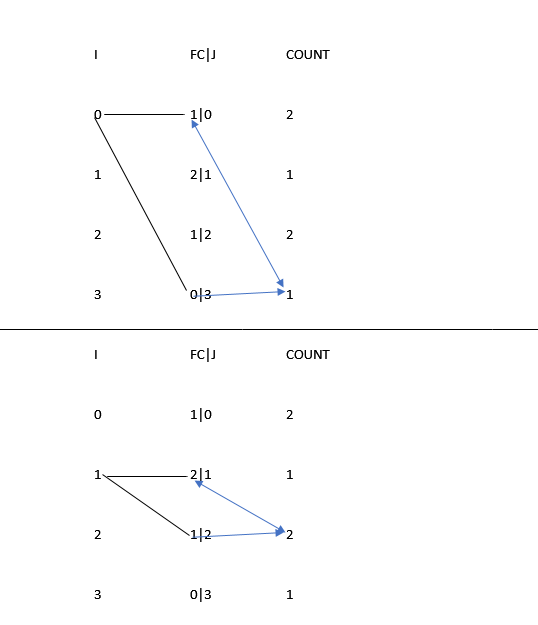
**TITLE: ASSIGNMENT 3**

**COURSE: CSC235**

**LINK: https://github.com/hamidusodiq/csc235---assignment-3**

**PSEUDOCODE TO SOLVE THE SELF DESCRIBING NUMBER PROBLEM**

**IMAGE PATTERN TO SOLVE PROBLEM**



* Read the file with fopen()
* create array countNo to store the number of times each digit appears
* create array intFileContent to store the integer equivalent of the file content
* read the first line of the file as number of test cases
* create an array testCases to store the number of test cases
* store the number of test cases into testCases array
* create an array fileContent to store each case value as string
* Check if file is valid. If not, print an error message and exit.
* convert number of testCases to int
* check if number of testCases is greater than 1,else throw an error
* while each test case is truthy:

countAppearance function:

* loop(i)-through the fileContent
  + initialize a count variable to 1 that represents the least appearance of each "string digit"
  + store the count of that variable in the countNo array at index of i
  + loop(ii): loop through the fileContent
  + if value of fileContent of loop1 equals that of loop 2:
    - value of loop appears again, hence increase count and update array of that index
    - else retain count to be last value recorded

checkDescribing function:

* Create an array(boolArray) to hold "T" or "F" based on value of number array equaling current index
* insert the string of digits as integers in intFileContent
* Loop(iii) through fileContent using i variable:
  + initialize all i values of boolArray[i] to T
  + loop4 though fileContent using j
  + Check if value of intFileContent at index j is equal to index of I:
    - IF true, check if value of intFileContent at index i = value of countNo at index j
      * If true, the current digit fulfils self-description, hence initialize the current index at boolArray to be true;
      * If false, the whole integer does not fulfil being self-descriptive. Hence print "Not self-descriptive”. Then return
* If all digits have been looped through and each fulfills self-description, then the whole integer is self-descriptive. Hence print self-descriptive!
* Close file!

**PSEUDOCODE TO SOLVE THE DUPLICATE FILE PROBLEM**

* Read the file using fopen
* Check if file is valid. If not, print an error message and exit
* Declare variable testCases to store the number of test cases
* Declare array fileValues to store the values in the file
* initialize a counter variable i=0 to store the number of lines in the file
* read the first line of the file as the number of test cases and store in testCases variable using fscanf
* loop through the rest of the file and determine the number of values by increasing the counter variable i
* loop through the file again and store each values into the fileValues Array created
* declare an array stringValues to store only values which are not digits in the files
* loop through the fileValues array
  + if current value is a digit(variable k), it represents the number of testCases within each test case
  + create variable y = 0 to keep track of the next values after condition above is met
  + loop through the next fileValues[K] lines
    - store each line value in the stringValues array
    - At end of loop, loop through the stringValues array and run the chekcStringEqual functions for the next two line values in each loop

checkstringEqual function:

* declare two array of characters splitString1 and splitString2 to store the next two values in the loop
* copy and split the next two values using strcpy and strtok into the splitString1 and splitString2 array to get the non-digit part of the file
* compare the split versions
* if not equal : print the last part of the two original strings which is the index part of the file in the order of which is smaller