
Article

[Stuart Byrne](#) · Apr 3, 2019 5m read

Belated - Advent of Code 2018 Day 2

Continuing on from my last post, I am posting my succesful output for the Day 2 Part 1 below.

A full explanation of Day 2 part 1 can be found at: <https://adventofcode.com/2018/day/2> and the input can be found at: <https://adventofcode.com/2018/day/2/input>

A brief description of this exercise is that you receive a sequence of strings.

You have to count the if any duplicate or triplicate chars exist in the string. Multiples will exist, but if you find multiple lduplicates / triplicates you will count them once.

At the end of sequence of strings you have to SUM the no. of duplicate / triplicate chars and multiply them together to get a simple check sum value.

I created 2 methods, one to import the data and the second to analyse the strings and populate the duplicate count and triple count properties.

Please let me know of alternative ways of dealing with this, as it always helps to be 'enlightened' of tackling an issue in a different way.

I do realise that I could do this in one method, but it made it simpler to split it in my mind.

Here is the code I tried below:

```
ClassMethod Import(dir As %String = "C:\Users\kanga\OneDrive\Documents\Advent of Code
\2018\Day2_Input.txt")
{
    //Open my text file and check it exists
    IF ##class(%File).Exists(dir) '=1 {
        write !, "Unable to Open: "_dir
        QUIT}
    ELSE { write "File: "_dir_" is open"}

    set file = ##class(%Stream.FileCharacter).%New()
    set sc = file.LinkToFile(dir)
    WHILE ('file.AtEnd) {
        //create incrememnt for going through each line of text from file
        //read line of text from file
        set line = file.ReadLine()
        set nline = $INCREMENT(nline)
        //if line exists loop through each line until no more lines
        IF (nline >= 1) & (line '= "") {
            write !, "nline loop started"
            //read line of text from file
            //set line = file.ReadLine()
            set row = ##class(AOC2018.Day2).%New()
```

```

    // get length of line
    set lline = $LENGTH(line)
    write !, line
    For i=1:1:lline {
        set row.WholeLine = line
        set readvalue = $EXTRACT(line,i,i)
        write !, "Current readvale is: "_readvalue_" i is currently: "_i
    }
    set status = row.%Save()
    }
    //set status = row.%Save()
//write status
}
Write !, "Import Complete"
}

/// Processes the input from Importv1 and looks to find duplicate values in the string
ClassMethod Part1()
{
    //initialising variables
    set rowcount = ""
    set line = ""
    set doublecount = ""
    set triplecount = ""
    set CheckSum = ""
    // Get the highest row number in the table WDA
    &sql(SELECT TOP 1 ID INTO :rowcount FROM AOC2018.Day2 ORDER BY ID DESC)
    Write !, "Row count is: "_rowcount
    // Loop through each row in the table
    FOR i=1:1:rowcount{
        //reset doublecount and triplecount for each row
        set doublecount = ""
        set triplecount = ""
        set duplicatecharlist = ""
        &sql(SELECT WholeLine INTO :line FROM AOC2018.Day2 WHERE ID = :i)
        write !, line
        set linelength = $LENGTH(line)
        set currentrow = ##class(AOC2018.Day2).%OpenId(i)
        For char=1:1:linelength{
            set firstrepeatchar = ""
            set Secondrepeatchar = ""
            /// get current character being looked for
            set currentchar = $EXTRACT(line,char,char)
            write !, "CurrentChar is: "_currentchar
            ///use find to see if current char exists in the rest of the string
            // write an IF statement to check if currentchar has already been identified as a duplicate.
            IF $LISTFIND(duplicatecharlist,currentchar) {
            }
            ELSE{
                set firstrepeatchar = $FIND(line,currentchar,char+1)
                // IF we find a repeat findrepeatchar will be >0
                IF firstrepeatchar >= 1 {
                    set duplicatecharlist = $LB(currentchar)
                    write !, "Duplicatechar is: "_currentchar
                    write !, "Current duplicate char is: "_currentchar
                    //After finding first repeat, increment doublecount by 1
                    set doublecount = $INCREMENT(doublecount)
                }
            }
        }
    }
}

```

```

//INSERT a statemnet to build a list of characters that are dups and use to stop counting dups twice.
IF firstrepeatchar-1 < linelength {
//look for second repeat of currentchar in string
set Secondrepeatchar = $FIND(line,currentchar,firstrepeatchar)
IF Secondrepeatchar >= 1{
write !, "Current triplicate char is: "_$EXTRACT(line,Secondrepeatchar-1,Secondrepeatchar-1)
//After finding second repeat, increment triplecount by 1
set triplecount = $INCREMENT(triplecount)
//reduce doublecount by 1 as currnet char is a triplicate
write !, "Double count pre-reduction: "_doublecount
set doublecount = doublecount-1
write !, "Double count post-reduction for triplicate: "_doublecount
}
}
}
IF doublecount > 1 {
set doublecount = 1
}
IF triplecount > 1{
set triplecount = 1
}
}
write !, "For Row: "_i_". We found: "_doublecount_" duplicates & "_triplecount_" triplicates"
set currrentrow.DuplicateCount = doublecount
set currrentrow.TriplicateCount = triplecount
set status = currrentrow.%Save()
write !, "Save status is: "_status
}
}
//reset sum doublecount and tripcount variable so can be used for generating checksum outside of loops
set doublecount = ""
set triplecount = ""
&sql(SELECT SUM(DuplicateCount) INTO :doublecount FROM AOC2018.Day2)
&sql(SELECT SUM(TriplicateCount) INTO :triplecount FROM AOC2018.Day2)
write !, "DoubleCount is: "_doublecount_" TripleCount is: "_triplecount
set CheckSum = doublecount*triplecount
write !, "Final Checksum is: "_CheckSum
}

```

[#Contest](#) [#ObjectScript](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/belated-advent-code-2018-day-2>