Text Processing in Linux - the 'uniq' command - #4

Problem Statement

Introduction and References

In linux, the most vanilla version of 'uniq' eliminates consecutive repetitions of a line when a text file is piped through it.

Plain Uniq

For instance, if this is the file **test.txt**

```
00
00
01
01
00
00
02
02
```

This is the output on passing it through the 'uniq' command, either via pipes or as input via STDIN.

```
Command:  uniq < test.txt

00
01
00
02
```

Since the first two lines of the original file are the same (00) and same for the next two (01) again followed by two repetitions of 00 and two repetitions of 02 - the 'uniq' command replaces consecutive repetitions by only one line in each case.

Uniq with counts

```
uniq -c < test.txt
```

This one also indicates the count of repetitions for each of the lines it collapses.

If this is the test file (say, testCounts.txt)

```
00
00
01
01
00
00
02
02
03
aa
aa
aa
```

uniq -c < input00.txt

```
2 00
2 01
2 00
2 02
1 03
3 aa
```

The first number is the **count** of the number of repeated occurrences in the original file.

Printing only duplicate lines

The '-d' option prints only those lines which are followed by one or more repetitions immediately after them.

```
uniq -d < testCounts.txt
```

OR

```
cat testCounts.txt | uniq -d
```

OR

```
uniq -d testCounts.txt
```

Printing only unique lines

The '-u' option prints only those lines which are succeeded and preceded by different lines.

```
uniq -u < testCounts.txt
```

OR

```
cat testCounts.txt | uniq -u
```

OR

```
uniq -u testCounts.txt
```

It is also possible to

- limit comparison only to the first N characters (using the -w option)
- avoid comparing first N characters (using the -s option)
- ignore variations in case between lines (the -i option)
- avoid comparing the first N fields using the -f option.
  (This may be useful while processing TSV files when you'd like to ignore the first column, if it has serial numbers.)

You might find these examples interesting and useful.

Current Task

Given a text file, display only those lines which are **not** followed or preceded by identical replications.

Sample Input

```
A00
a00
01
01
00
00
02
02
A00
03
aa
aa
aa
```

Sample Output

```
A00
a00
A00
03
```

Explanation

The comparison is case sensitive, so the first instance of "A00" and "a00" are considered different, hence unique.
The next instance of A00 is succeeded and preceded by different lines, so that is also included in the output.
The same holds true for 03 - it is succeeded and preceded by different lines, so that is also included in the output.