

'Grep' - B

An Introduction to Grep

Grep is a multi-purpose search tool, which is used to find specified strings or regular expressions. A variety of options exist, which make it possible to use the command in several different ways and to handle many different situations. For example, one might opt for case-insensitive search, or to display **only** the fragment matching the specified search pattern, or to display only the line number of an input file where the specified string or regular expression has been found.

More details about common examples of grep usage may be read [here](#).

Before using **grep** it is recommended that one should become familiar with regular expressions, to be able to harness the command to its fullest.

Recommended References

- [15 Practical Grep Command Examples](#)
- [TLDP Examples for Grep](#)
- [Grep Regular Expressions](#)
- [Grep Regular Expressions on the GNU site](#)

Current Task

Given an input file, with **N** credit card numbers, each in a new line, your task is to grep out and output only those credit card numbers which have two or more consecutive occurrences of the same digit (which may be separated by a space, if they are in different segments). Assume that the credit card numbers will have 4 space separated segments with 4 digits each.

If the credit card number is 1434 5678 9101 1234, there are two consecutive instances of 1 (though) as highlighted in box brackets: 1434 5678 910[1] [1]234

Here are some credit card numbers where consecutively repeated digits have been highlighted in box brackets. The last case does not have any repeated digits: 1234 5678 910[1] [1]234
2[9][9][9] 5178 9101 [2][2]34
[9][9][9][9] 5628 920[1] [1]232
8482 3678 9102 1232

Input Format

N credit card numbers. Assume that the credit card numbers will have 4 space separated segments with 4 digits each.

Constraints

$$1 \leq N \leq 20$$

However, the value of **N** does not matter while writing your command.

Output Format

Display the required lines after filtering with grep, without any changes to their relative ordering in the input file.

Sample Input

```
1234 5678 9101 1234
```

```
2999 5178 9101 2234
9999 5628 9201 1232
8482 3678 9102 1232
```

Sample Output

```
1234 5678 9101 1234
2999 5178 9101 2234
9999 5628 9201 1232
```

Explanation

Consecutively repeated digits have been highlighted in box brackets. The last case does not have any repeated digits: 1234 5678 910[1] [1]234

2[9][9][9] 5178 9101 [2][2]34

[9][9][9][9] 5628 920[1] [1]232

8482 3678 9102 1232