Sed is a popular utility which enables quick parsing and transformation of text.

Here are some very simple examples of **sed** in action.

Substitute the first occurrence of 'editor' with 'tool'.

```
`$:~/user/bash/grep/grep1$` echo "My favorite programming editor is Emacs. Another editor I like is Vim." | sed -e s/editor/tool/
My favorite programming tool is Emacs. Another editor I like is Vim.
```

Substitute all the occurrences of 'editor' with 'tool'.

```
`$:~/user/bash/grep/grep1$` echo "My favorite programming editor is Emacs. Another editor I like is Vim." | sed -e s/editor/tool/g
My favorite programming tool is Emacs. Another tool I like is Vim.
```

Substitute the second occurrence of 'editor' with 'tool'.

```
`$:~/user/bash/grep/grep1$` echo "My favorite programming editor is Emacs. Another editor I like is Vim." | sed -e s/editor/tool/2
My favorite programming editor is Emacs. Another tool I like is Vim.
```

Highlight all the occurrences of 'editor' by wrapping them up in brace brackets.

```
`$:~/user/bash/grep/grep1$` echo "My favorite programming editor is Emacs. Another editor I like is Vim." | sed -e s/editor/{\&}/g
My favorite programming {editor} is Emacs. Another {editor} I like is Vim.
```

Some references for learning about **sed** have been included:

Sed - An Introduction and a tutorial

The TLDP Guide

Some Practical Examples

Task

Given an input file, with **N** credit card numbers, each in a new line, your task is to **reverse the ordering of segments** in each credit card number. Assume that the credit card numbers will have 4 space separated segments with 4 digits each.

If the original credit card number is 1434 5678 9101 1234, transform it to 1234 9101 5678 1434.

Useful References: This particular page on StackOverflow has a relevant example about sed, groups and backreferences. Here's a detailed tutorial covering groups and backreferences.

Input Format

N credit card numbers, each in a new line, credit card numbers will have 4 space separated segments with 4 digits each.

Constraints

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

Output Format

N lines, each containing a credit card number with the ordering of its segments reversed.

Sample Input

```
1234 5678 9101 1234
2999 5178 9101 2234
9999 5628 9201 1232
8888 3678 9101 1232
```

Sample Output

```
1234 9101 5678 1234
2234 9101 5178 2999
1232 9201 5628 9999
1232 9101 3678 8888
```

Explanation

The order of the four segments in the (input) credit card numbers have been reversed.

Questions

If you need any clarification, please contact us.