# 'Sed' command #4



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'.

`\$:~/hackerrank/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'.

`\$:~/hackerrank/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'.

`\$:~/hackerrank/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.

` $$:\sim\hookerrank/bash/grep/grep1$$ ` 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 mask the first 12 digits of each of the credit card numbers with the asterisk sign. Assume that the credit card numbers will have 4 space separated segments with 4 digits each.

If the original credit card number is 1234 5678 9101 1234, transform it to \*\* \* \*\*\* 1234

**Useful References:** This particular page on StackOverflow has a slighly modified version of the same task, where the solution involves backreferences. Here's a detailed tutorial covering groups and backreferences.

### **Input Format**

**N** credit card numbers, each in a new line, your task is to mask the first 12 digits of each of the credit card numbers with the asterisk sign. Assume that the credit card numbers will have 4 space separated segments with 4 digits each.

#### **Constraints**

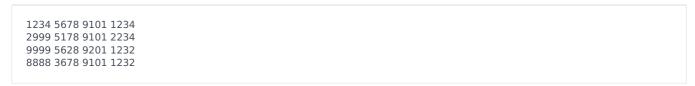
1<=N<=20

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

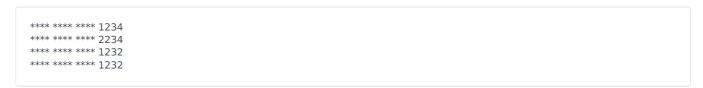
#### **Output Format**

**N** lines, each containing the masked version of the credit card number in the corresponding line of the input file.

## **Sample Input**



## **Sample Output**



## **Explanation**

The digits of the first 3 segments have been masked in each of the credit card numbers.