Week 8, Code with us

Problem-1

A software company has published some best practices for writing the code. One of the best practices mentioned is that no line in your code should exceed 50 characters in total including all types of characters or spaces.

Write a bash script using sed that prints the names of all .c files that contain one or more lines with a length of more than 50 characters(as specified above).

Hint: Use q [exit code]

Solution

```
for file in *.c; do
  sed -nE '/.{51,}/ q 51' $file || echo $file
done
```

Problem-2

Write a command using sed to print the contents of file in the format `<line number>:<contents>`.

Solution

```
sed '=' $filename | sed 'N; s/\n/:/'
```

Problem-3

The command `cut -d " " -f 9 lsoutput` is executed to extract the size of the file. But the `lsoutput` contains multiple spaces between fields and for some reason we cannot change the cut command. Your task is to pre-process the file `lsoutput` to work with the cut command.

Hint: Use the -i option in sed to do the modification in the file.

Solution

```
sed -i -E "s/ {2,}/ /g" lsoutput
```

Problem-4

In python, the multiline comments can be bounded by either `'''` or `"""`.

Write a sed script `script.sed` that converts all the comments inside the triple quotes to single-line comment (preceded by #) and remove the lines inside the triple quotes that do not have any contents.

Sample Input:

```
print(1)
"""
This is a comment 1
"""
print(2)
"""This is a comment 2
"""
print(3)
"""
This is a comment 3"""
print(4)
"""This is a comment 4"""
print(5)
"""This is a comment 5"""
print(6)
```

Sample Output

```
print(1)
#This is a comment 1
print(2)
#This is a comment 2
print(3)
#This is a comment 3
print(4)
#This is a comment 4
print(5)
#This is a comment 5
print(6)
```

Solution

```
# To remove inline triple quotes
s/['"]\{3\}\(.*\)['"]\{3\}/#\1/;

# To remove multi-line triple quotes
/['"]\{3\}/, /['"]\{3\}/ s/^/#/; s/['"]\{3\}//g; /^#$/d
```

Problem-5

Write a sed script `script.sed` to extract the content inside every `<td>` tags. Print all the non-empty extracted contents with leading and trail spaces trimmed in the source order.

Note: Add more options to sed execution in main.sh if required.

Sample Input

```
<html>
<head>
    <title>Example script for sed</title>
</head>

<body>
    <h1>Welcome to sed programming</h1>
    <p>sed is a steam editor known for manipulation of text.<br>
        sed can manipulate the text in the pipeline and can be used alond with other
commands as well</p>
    <b>Frequently used options with sed</b>
    <table>
        <tr>
            <td> -n, --quiet, --silent </td>
            <td> suppress automatic printing of pattern space </td>
        </tr>
        <tr>
            <td> -e script, --expression=script </td>
            <td> add the script to the commands to be executed </td>
        </tr>
        <tr>
            <td> -f script-file, --file=script-file </td>
            <td> add the contents of script-file to the commands to be executed </td>
        </tr>
        <tr>
            <td> -i[SUFFIX], --in-place[=SUFFIX] </td>
            <td> edit files in place (makes backup if SUFFIX supplied) </td>
        </tr>
        <tr>
            <td> -l N, --line-length=N </td>
            <td> specify the desired line-wrap length for the `l' command </td>
        </tr>
        <tr>
            <td> -E, -r, --regexp-extended </td>
            <td> use extended regular expressions in the script (for portability use
POSIX -E). </td>
        </tr>
    </table>
</body>
</html>
```

Sample Output

```
-n, --quiet, --silent
suppress automatic printing of pattern space
-e script, --expression=script
add the script to the commands to be executed
-f script-file, --file=script-file
add the contents of script-file to the commands to be executed
-i[SUFFIX], --in-place[=SUFFIX]
edit files in place (makes backup if SUFFIX supplied)
-l N, --line-length=N
specify the desired line-wrap length for the `l' command
-E, -r, --regexp-extended
use extended regular expressions in the script (for portability use POSIX -E).
```

Solution

```
#!/usr/bin/sed -f

/<td>/, /<\/td>/ {
  s/<td>//
  s/<\/td>//
  s/^[[:space:]]*//g
  s/[[:space:]]*$//g
  /^$/d
  p
}
```