

St. Francis Institute of Technology, Mumbai-400 103.
Department of Information Technology

A.Y. 2023-2024

Class: SE-ITA/B, Semester: IV

Subject: **UNIX LAB**

Experiment – 9A: grep/ egrep script programming.

1. **Aim:** To study and implement grep/ egrep script programming.
2. **Objectives:**
 - To understand grep and egrep scripting.
 - To understand the use of grep filter.
3. **Outcomes:** After study of this experiment, the student will be able to
 - Develop shell scripts using grep and egrep filter.
 - Perform pattern matching using grep scripts.
4. **Prerequisite:** Filters, shell scripts.
5. **Requirements:** Personal Computer, Ubuntu OS, Text Editor, LibreOffice.
6. **Pre-Experiment Exercise:**

Brief Theory:

Grep command

The grep command is a pattern matching tool. The grep filter scans its input for a particular pattern of characters, and displays all lines or the filenames that contain that pattern. The pattern that is searched is referred to as the regular expression. grep stands for globally search for regular expression and print out.

Grep is also a filter, so its output can be redirected to a file or to another command.

Syntax:

grep [options] pattern [files]

Grep options description

1. **-c:** This prints only a count of the lines that match a pattern.
2. **-h:** Display the matched lines, but do not display the filenames.
3. **-i:** Ignores, case for matching.
4. **-l:** Displays list of a filenames only.
5. **-n:** Display the matched lines and their line numbers.
6. **-r:** Searches the pattern recursively.
7. **-v:** This prints out all the lines that do not matches the pattern.
8. **-e exp:** Specifies expression with this option. Can use multiple times.
9. **-f file:** Takes patterns from file, one per line.
10. **-E:** Treats pattern as an extended regular expression (ERE).
11. **-F:** Treats pattern as a set of multiple fixed strings.
12. **-w:** Match whole word
13. **-o:** Print only the matched parts of a matching line, with each such part on a separate output line.

Egrep

On Unix-like operating systems, the egrep command searches for a text pattern, using extended regular expressions to perform the match. Running egrep is equivalent to running grep with the -E option.

Syntax:

grep [options] pattern [files].

7. Laboratory Exercise

A. Procedure

1. Write a grep script to find the number of matched characters, words and lines in a file.
2. Write a grep script to find the number of characters, words and lines in a file.
3. Write an egrep script to display list of files starting with particular letter in the directory.

B. Result/Program code Screenshots

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

1. Write a shell program using grep to count the number of files in a Directory.

C. Conclusion:

1. Write what was performed in the experiment.
2. Mention few applications of what was studied.
3. Write the significance of the topic studied in the experiment.

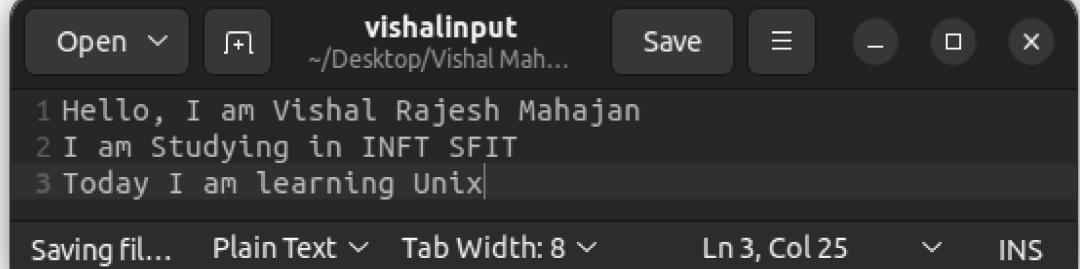
D. References:

1. Yashwant Kanetkar, UNIX Shell Programming, BPB Publications.
2. Sumitabha Das, UNIX Concepts and Applications, 3rd Ed., Tata McGraw Hill.
3. <https://www.geeksforgeeks.org/grep-command-in-unixlinux/>.

Name: Vishal Rajesh Mahajan
Class: SE IT A

Exp: 9A
Roll No: 63

EXP 9A: grep / egrep script programming

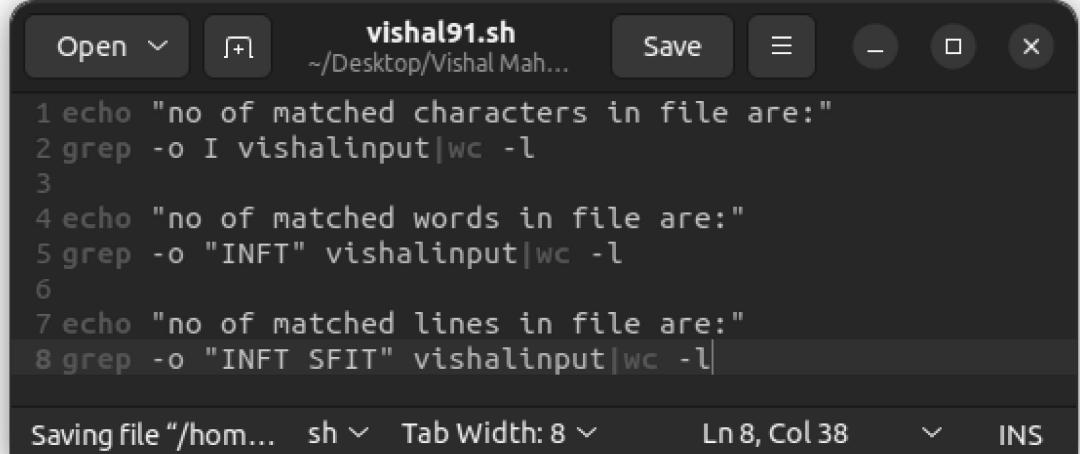


```
vishalininput
~/Desktop/Vishal Mah...
1 Hello, I am Vishal Rajesh Mahajan
2 I am Studying in INFT SFIT
3 Today I am learning Unix|
```

Saving fil... Plain Text ~ Tab Width: 8 ~ Ln 3, Col 25 ~ INS

Vishalininput text file

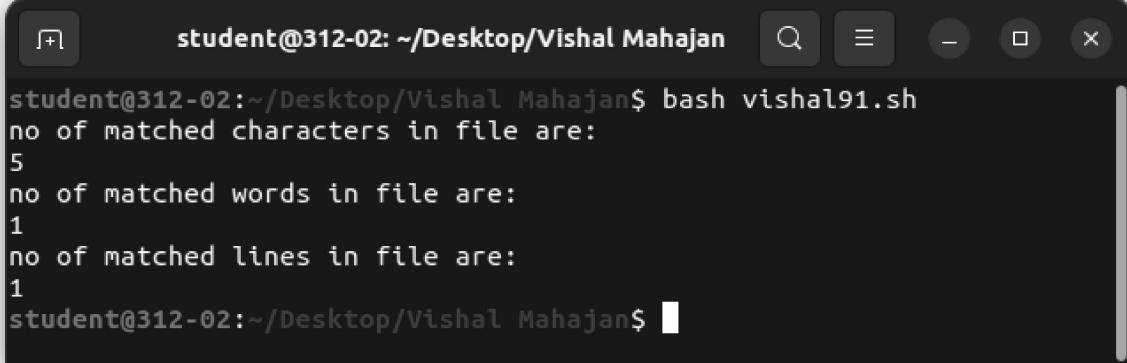
1. Write a grep script to find the number of matched characters, words, lines in a file



```
vishal91.sh
~/Desktop/Vishal Mah...
1 echo "no of matched characters in file are:"
2 grep -o I vishalininput|wc -l
3
4 echo "no of matched words in file are:"
5 grep -o "INFT" vishalininput|wc -l
6
7 echo "no of matched lines in file are:"
8 grep -o "INFT SFIT" vishalininput|wc -l|
```

Saving file "/hom... sh ~ Tab Width: 8 ~ Ln 8, Col 38 ~ INS

Grep script of Program 1



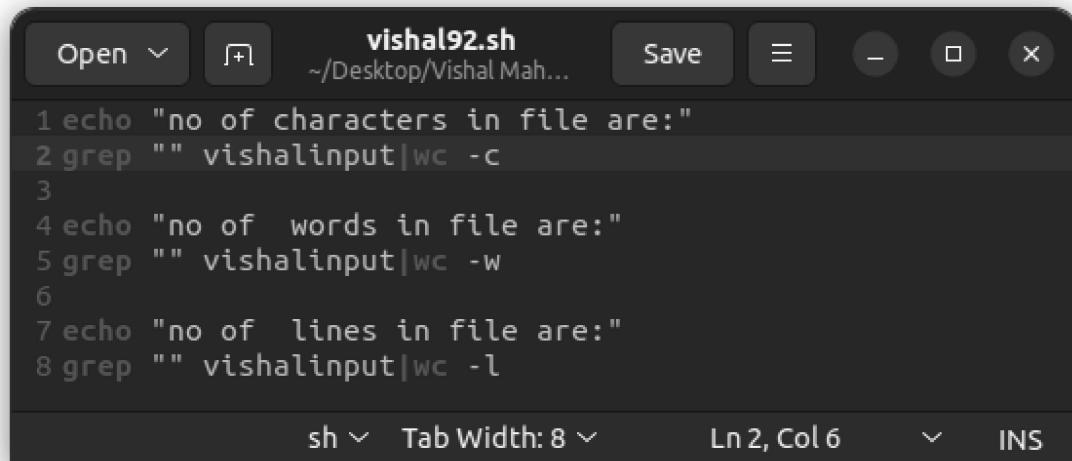
```
student@312-02: ~/Desktop/Vishal Mahajan$ bash vishal91.sh
no of matched characters in file are:
5
no of matched words in file are:
1
no of matched lines in file are:
1
student@312-02: ~/Desktop/Vishal Mahajan$
```

Output of Program 1

Name: Vishal Rajesh Mahajan
Class: SE IT A

Exp: 9A
Roll No: 63

2. Write a grep script to find the number of characters, words, lines in a file



```
vishal92.sh
~/Desktop/Vishal Mah...
Open Save
1 echo "no of characters in file are:"
2 grep "" vishalininput|wc -c
3
4 echo "no of words in file are:"
5 grep "" vishalininput|wc -w
6
7 echo "no of lines in file are:"
8 grep "" vishalininput|wc -l

sh Tab Width: 8 Ln 2, Col 6 INS
```

Grep script of Program 2



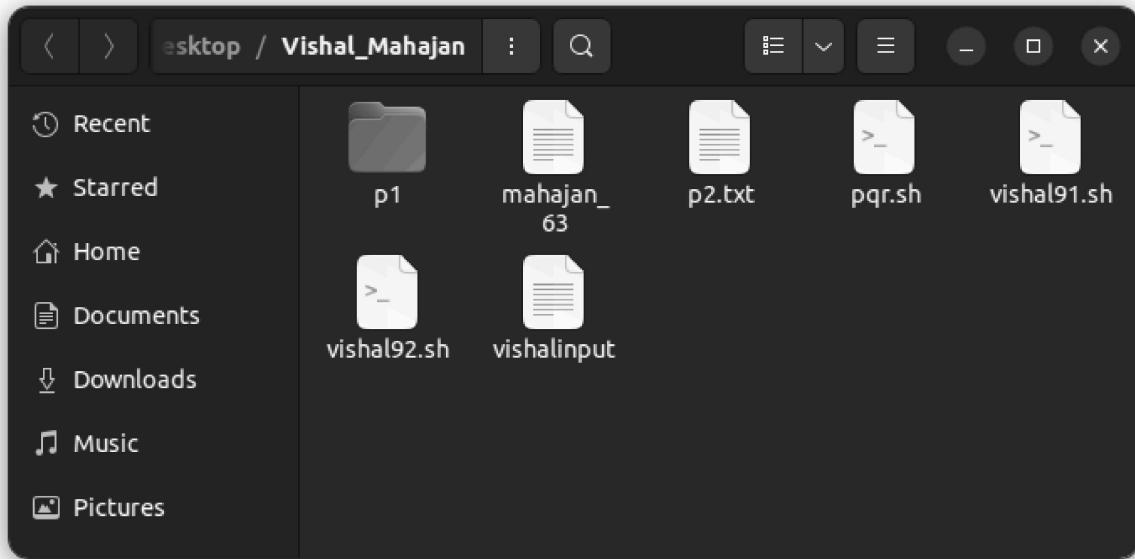
```
student@312-02: ~/Desktop/Vishal Mahajan$ bash vishal92.sh
no of characters in file are:
86
no of words in file are:
17
no of lines in file are:
3
student@312-02: ~/Desktop/Vishal Mahajan$
```

Output of Program 2

Name: Vishal Rajesh Mahajan
Class: SE IT A

Exp: 9A
Roll No: 63

3. Write a egrep script to display list of files starting with particular letter in the directory



Vishal_Mahajan Directory

```
1 #! /bin/bash
2 echo "enter the directory name:"
3 read dir
4 if [ -d $dir ]
5 then
6     echo " List of files in directory starting with p
    are : "
7     ls $dir | egrep "^\p" | wc -l
8 else
9     echo "Enter proper directory name"
10 fi
```

The screenshot shows a terminal window with the script 'vishal93.sh' open. The script is a Bash script that prompts the user for a directory name, reads it into the variable \$dir, checks if it's a directory using [-d], and then lists all files in that directory that start with the letter 'p' using egrep "^\p". The script then counts the number of files using wc -l. If the directory name is not provided or is not a directory, it prints an error message. The terminal window also shows the file path ~/Desktop/vishal93.sh and the current line and column information (Ln 1, Col 1).

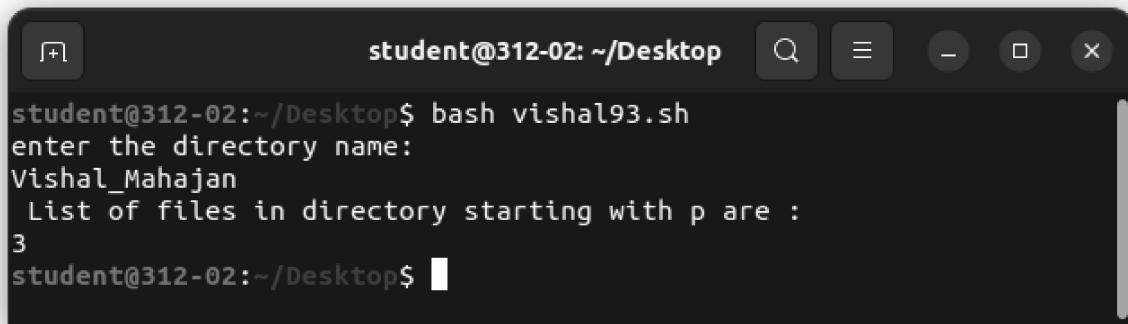
Script of Program 3

Name: Vishal Rajesh Mahajan

Exp: 9A

Class: SE IT A

Roll No: 63



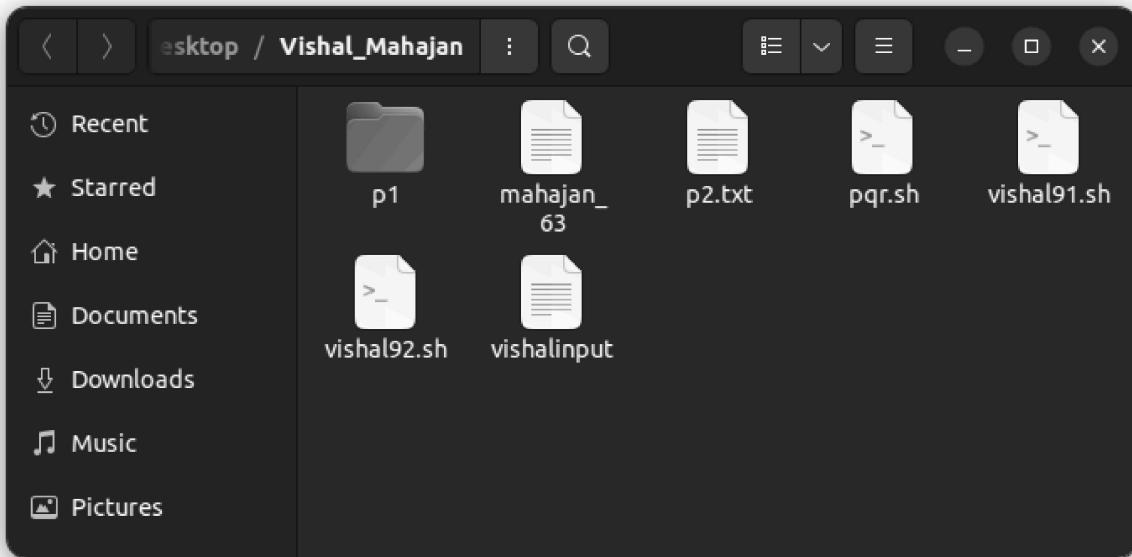
A screenshot of a terminal window titled "student@312-02: ~/Desktop". The window contains the following text:

```
student@312-02:~/Desktop$ bash vishal93.sh
enter the directory name:
Vishal_Mahajan
List of files in directory starting with p are :
3
student@312-02:~/Desktop$
```

Output of Program 3

POST-EXPERIMENT

1. Write a shell program using grep to count the number of files in a directory



Vishal_Mahajan Directory

```
1 #! /bin/bash
2 echo "enter the directory name:"
3 read dir
4 if [ -d $dir ]
5 then
6     echo " Total Number of file in a directory is| "
7     ls $dir | egrep "" | wc -l
8 else
9     echo "Enter proper directory name"
10 fi
```

Saving file "/hom... sh ~/Desktop Ln 6, Col 54 INS

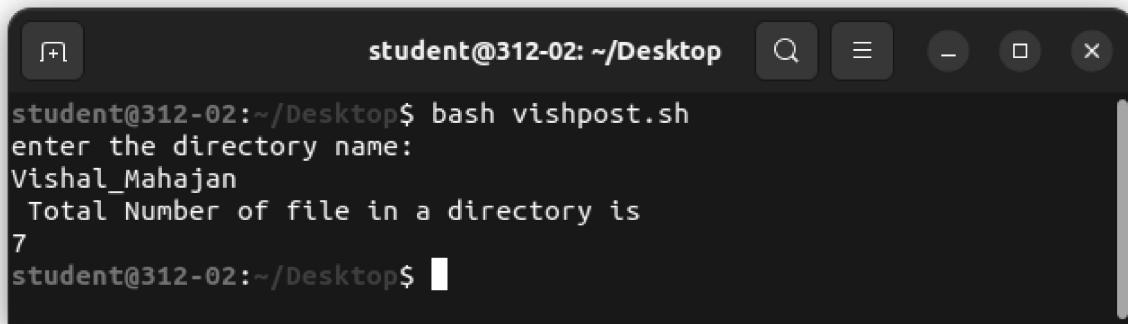
Script of Post EXP program

Name: Vishal Rajesh Mahajan

Exp: 9A

Class: SE IT A

Roll No: 63



A screenshot of a terminal window titled "student@312-02: ~/Desktop". The window contains the following text:

```
student@312-02:~/Desktop$ bash vishpost.sh
enter the directory name:
Vishal_Mahajan
Total Number of file in a directory is
7
student@312-02:~/Desktop$
```

Output of Post Experiment Program

St. Francis Institute of Technology, Mumbai-400 103.
Department of Information Technology

A.Y. 2023-2024

Class: SE-ITA/B, Semester: IV

Subject: **UNIX LAB**

Experiment – 9B: Advanced filtering with AWK script programming.

1. **Aim:** To study and implement AWK script programming.

2. **Objectives:**

- To understand and implement AWK scripts.
- To understand the use of AWK scripts.

3. **Outcomes:** After study of this experiment, the student will be able to

- Understand and implement AWK scripts.
- Use AWK filter as a report formatting tool.

4. **Prerequisite:** Filters, shell scripts.

5. **Requirements:** Personal Computer, Ubuntu OS, Text Editor, LibreOffice.

6. **Pre-Experiment Exercise:**

Brief Theory:

AWK:

Awk is a scripting language used for **manipulating data and generating reports**. The awk command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Unlike other filters, it operates at the **field** level and can easily access, transform and format individual fields in a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then performs the associated actions.

Awk is named after its authors – Aho, Weinberger, and Kernighan.

Syntax: awk options 'selection _criteria {action }' input-file(s)

AWK workflow

AWK follows a simple workflow – Read, Execute, and Repeat. Figure1. depicts the workflow of AWK.

Read

AWK reads a line from the input stream (file, pipe, or stdin) and stores it in memory.

Execute

All AWK commands are applied sequentially on the input. By default, AWK execute commands on every line. We can restrict this by providing patterns.

Repeat

This process repeats until the file reaches its end.

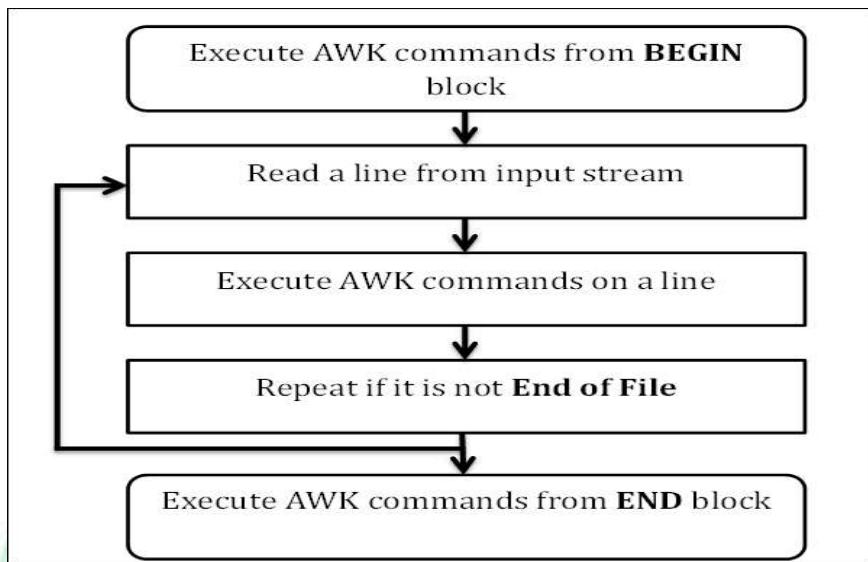


Figure 1. AWK Workflow.

7. Laboratory Exercise

A. Procedure

1. Write an AWK script to develop a Fibonacci series.
2. Write an AWK script to replace the Nth occurrence of a pattern.

B. Result/Program code Screenshots

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

1. Write an AWK script to print only certain columns from the input field.
2. Write an AWK script to display the pattern from a file.

C. Conclusion:

1. Write what was performed in the experiment.
2. Mention few applications of what was studied.
3. Write the significance of the topic studied in the experiment.

D. References:

1. Yashwant Kanetkar, **UNIX Shell Programming**, BPB Publications.
2. Sumitabha Das, **UNIX Concepts and Applications**, 3rd Ed., Tata McGraw Hill.
3. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/>.

EXP 9B: Advanced Filtering with AWK Script Programming.

1. Write a AWK Scripts to develop a Fibonacci Series

The screenshot shows a code editor window titled "vishal9B1.sh" located at "/Desktop/Vishal_Mah...". The script content is as follows:

```
1 awk 'BEGIN {
2     for (i=0;i<=10;i++)
3 {
4     if (i <= 1)
5     {
6         x=0;
7         y=1;
8         print i;
9     }
10    else
11    {
12        z= x + y;
13        print z;
14        x=y;
15        y=z;
16    }
17 }
18 }'
```

At the bottom of the editor, it says "Saving file "/hom... sh" Tab Width: 8" and "Ln 13, Col 30".

AWk Script of Fibonacci Series

The screenshot shows a terminal window titled "student@312-02: ~/Desktop/Vishal_Mahajan". The command "bash vishal9B1.sh" is run, and the output is displayed as follows:

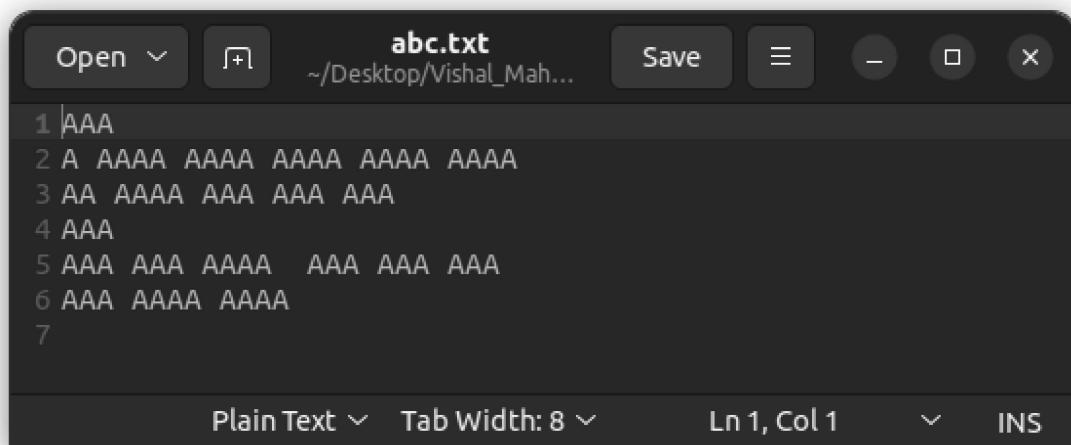
```
student@312-02:~/Desktop/Vishal_Mahajan$ bash vishal9B1.sh
0
1
1
2
3
5
8
13
21
34
55
student@312-02:~/Desktop/Vishal_Mahajan$
```

Output of AWk Script of Fibonacci Series

Name: Vishal Rajesh Mahajan
Class: SE IT A

Exp: 9B
Roll No: 63

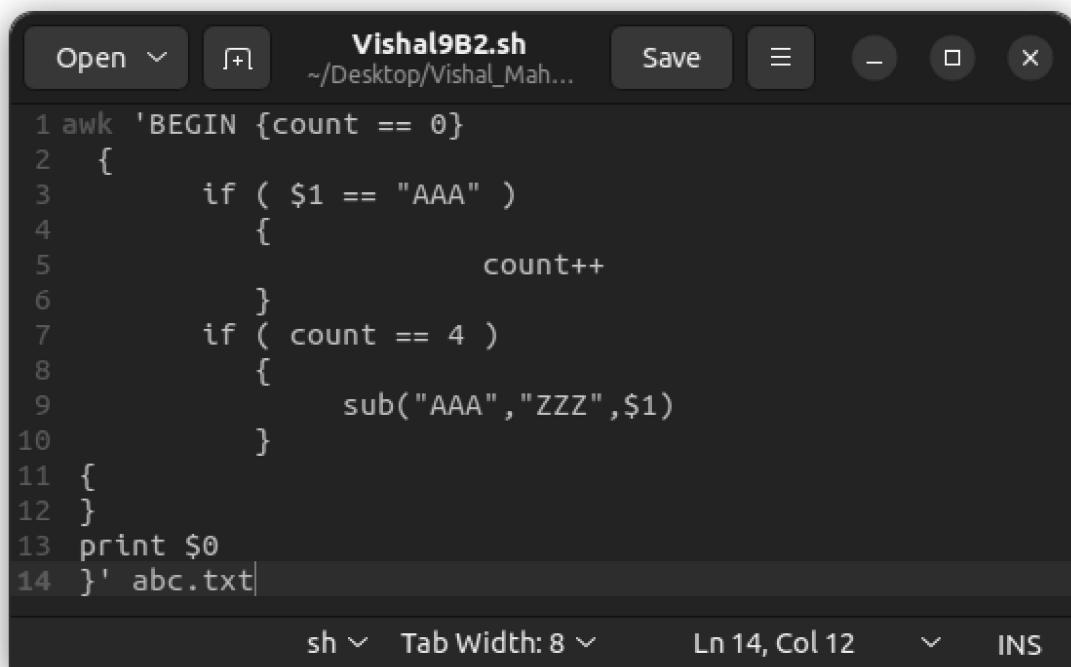
2. Write a AWK Scripts to replace the Nth occurrence of a pattern



The screenshot shows a text editor window titled "abc.txt". The content of the file is as follows:

```
1 AAA
2 A AAAAA AAAAA AAAAA AAAAA AAAAA
3 AA AAAAA AAA AAA AAA AAA
4 AAA
5 AAA AAA AAAAA AAA AAA AAA
6 AAA AAAAA AAAAA
7
```

The file is located at "/Desktop/Vishal_Mah...". The bottom status bar indicates "Plain Text" mode, a tab width of 8, and the current position is "Ln 1, Col 1".



The screenshot shows a text editor window titled "Vishal9B2.sh". The content of the file is an AWK script:

```
1 awk 'BEGIN {count == 0}
2 {
3     if ( $1 == "AAA" )
4     {
5         count++
6     }
7     if ( count == 4 )
8     {
9         sub("AAA","ZZZ",$1)
10    }
11 }
12 }
13 print $0
14 }' abc.txt
```

The file is located at "/Desktop/Vishal_Mah...". The bottom status bar indicates "sh" mode, a tab width of 8, and the current position is "Ln 14, Col 12".

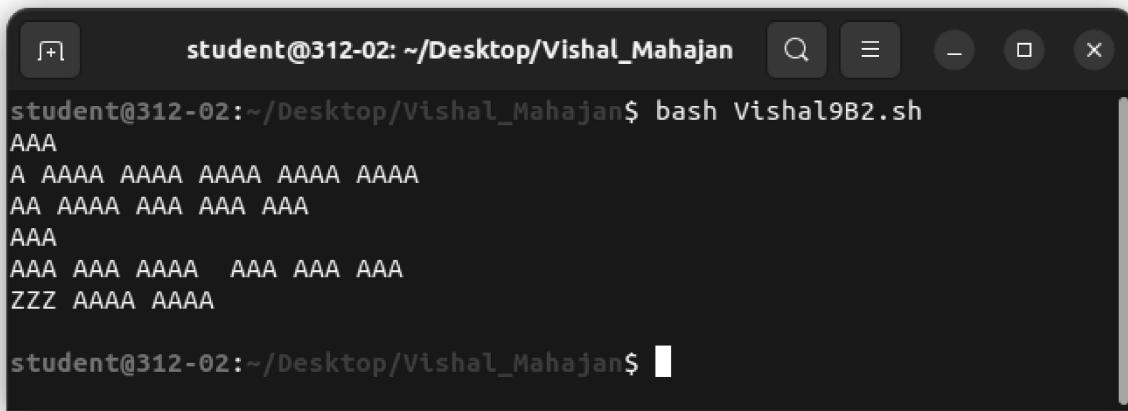
AWk Script of replacing Nth pattern

Name: Vishal Rajesh Mahajan

Exp: 9B

Class: SE IT A

Roll No: 63



A screenshot of a terminal window titled "student@312-02: ~/Desktop/Vishal_Mahajan". The window contains the following text:

```
student@312-02:~/Desktop/Vishal_Mahajan$ bash Vishal9B2.sh
AAA
A AAAA AAAA AAAA AAAA AAAA
AA AAAA AAA AAA AAA
AAA
AAA AAA AAAA AAA AAA AAA
ZZZ AAAA AAAA

student@312-02:~/Desktop/Vishal_Mahajan$
```

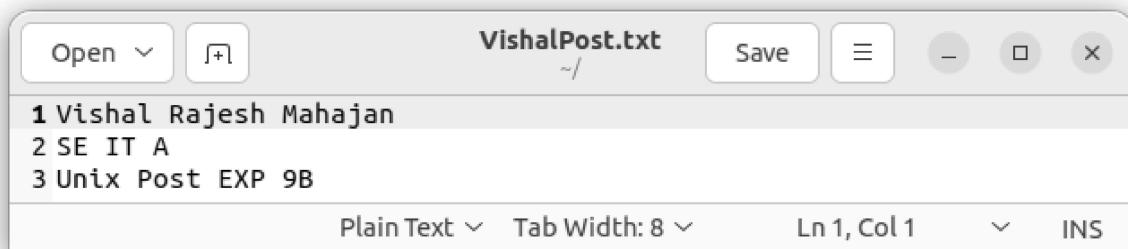
Output of AWk Script of replacing Nth pattern

Name: Vishal Rajesh Mahajan
Class: SE IT A

Exp: 9B
Roll No: 63

POST-EXPERIMENT EXERCISE:

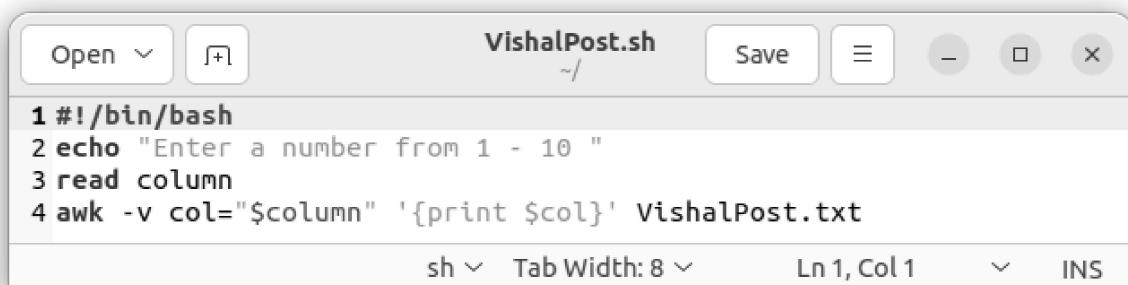
1. Write an AWK script to print only certain columns from the input field.



```
VishalPost.txt
1 Vishal Rajesh Mahajan
2 SE IT A
3 Unix Post EXP 9B
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

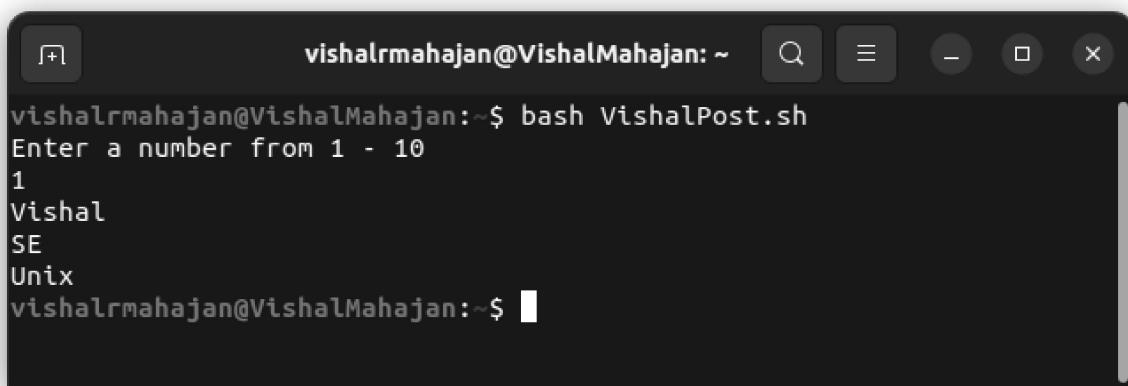
Text file of Post EXP 1



```
VishalPost.sh
#!/bin/bash
echo "Enter a number from 1 - 10"
read column
awk -v col="$column" '{print $col}' VishalPost.txt
```

sh ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

Awk Script of Post EXP 1



```
vishalrmahajan@VishalMahajan:~$ bash VishalPost.sh
Enter a number from 1 - 10
1
Vishal
SE
Unix
vishalrmahajan@VishalMahajan:~$
```

Output of Awk Script of Post EXP 1

Name: Vishal Rajesh Mahajan
Class: SE IT A

Exp: 9B
Roll No: 63

2. Write an AWK script to display the pattern from a file.

VishalPost2.txt

```
1 Vishal Rajesh Mahajan
2 Vishal| is studying in SFIT
3 Vishal roll number is 63
4
5 Unix Lab EXP 9B
6 Unix Lab Post EXP
7 Unix Lab is to learn Unix
8
```

Plain Text ▾ Tab Width: 8 ▾ Ln 2, Col 7 ▾ INS

Text file of Post EXP 2

VishalPost2.sh

```
1 echo "Enter a Pattern: "
2 read pattern
3 |
4 awk -v pat="$pattern" '$0 ~ pat' VishalPost2.txt
```

sh ▾ Tab Width: 8 ▾ Ln 3, Col 1 ▾ INS

Awk Script of Post EXP 2

```
vishalrmahajan@VishalMahajan:~$ bash VishalPost2.sh
Enter a Pattern:
Vishal
Vishal Rajesh Mahajan
Vishal is studying in SFIT
Vishal roll number is 63
vishalrmahajan@VishalMahajan:~$
```

Output of Awk Script of Post EXP 2