LAB 2: ADVANCED UNIX SHELL COMMANDS

Lab Exercises Solutions with Sample Outputs

1. Execute all the commands explained in this section and write the output.

Examples for each command:

- **grep**

```
$ echo -e "apple\napples\npineapple\nbanana\npear\npeach\norange" > fruitlist.txt

$ grep apple fruitlist.txt
apple
apples
pineapple
```

- **sort**

```
$ echo -e "pear\napple\nbanana\norange" > fruits.txt

$ sort fruits.txt
apple
banana
orange
pear

$ sort -r fruits.txt
pear
orange
banana
apple
```

- **wc**

```
$ wc fruits.txt
4 4 26 fruits.txt  # 4 lines, 4 words, 26 bytes

$ wc -l fruits.txt
4 fruits.txt

$ wc -w fruits.txt
4 fruits.txt
```

```
$ wc -c fruits.txt
26 fruits.txt
```

- **cut**

```
$ echo -e "one:two:three\nfour:five:six" > data.txt

$ cut -d ':' -f 2 data.txt
two
five

$ cut -c 1-3 data.txt
one
fou
```

- **sed**

```
$ cat data.txt
one:two:three
four:five:six

$ sed -e 's/two/TWO/' data.txt
one:TWO:three
four:five:six

$ sed -e '/^one/ s/one/ONE/' data.txt
ONE:two:three
four:five:six

$ sed -n -e '/^four/ p' data.txt
four:five:six

$ sed -e '/^one/d' data.txt
four:five:six
```

- **tr**

```
$ echo "hello world" | tr '[a-z]' '[A-Z]'
HELLO WORLD

$ echo "aaabbbccc" | tr -s 'a'  # compress consecutive 'a' to one 'a'
abc

$ echo "hello123" | tr -d '0-9' # delete digits
hello
```

2. Write grep commands to do the following activities:

- **a. To select the lines from a file that have exactly two characters**:

```
$ echo -e "a\nab\nabc\nabcd\nxy\n12\n1" > test.txt

$ grep -E '^.{2}$' test.txt
ab
xy
12
```

- **b. To select the lines that start with an uppercase letter:**

```
$ echo -e "Apple\nbanana\nPear\ngrape\nOrange" > test.txt

$ grep '^[A-Z]' test.txt
Apple
Pear
Orange
```

- **c. To select the lines that end with a period:**

```
$ echo -e "Hello world.\nHello again\nThis is a test." > test.txt

$ grep '\.$' test.txt
Hello world.
This is a test.
```

- **d. To select lines that have one or more blank spaces:**

```
$ echo -e "nospaces\none space\nmultiple spaces here" > test.txt

$ grep '[[:space:]]' test.txt
one space
multiple spaces here
```

- **e. To select lines that have digits and redirect them to another file:**

```
$ echo -e "line1\nline two\n3rd line\nfourth5\nsix" > test.txt

$ grep '[0-9]' test.txt > digits_lines.txt

$ cat digits_lines.txt
line1
3rd line
```

```
fourth5
```

3. Create studentInformation.txt file (using vi or echo) with at least 10 rows:

```
$ cat > studentInformation.txt << EOF
RegistrationNo:Name:Department:Branch:Section:Sub1:Sub2:Sub3
1234:XYZ:ICT:CCE:A:80:60:70
2345:ABC:ICT:IT:B:90:75:85
3456:PQR:ECE:ECE:A:70:80:75
4567:LMN:ICT:IT:A:88:92:78
5678:DEF:CSE:CSE:B:85:65:95
6789:GHI:ICT:CCE:B:60:70:80
7890:JKL:ECE:ECE:A:75:85:90
8901:MNO:ICT:IT:B:82:77:88
9012:PST:CSE:CSE:A:90:80:70
0123:QRS:ICT:CCE:B:85:90:80
EOF
```

i) Display the number students (count) belonging to ICT department:

```
$ grep -c ':ICT:' studentInformation.txt
5
```

ii) Replace all occurrences of "IT" branch with "Information Technology" and save output to ITStudents.txt:

```
$ sed 's/:IT:/:Information Technology:/g' studentInformation.txt > ITStudents.txt

$ head -5 ITStudents.txt
RegistrationNo:Name:Department:Branch:Section:Sub1:Sub2:Sub3
1234:XYZ:ICT:CCE:A:80:60:70
2345:ABC:ICT:Information Technology:B:90:75:85
3456:PQR:ECE:ECE:A:70:80:75
4567:LMN:ICT:Information Technology:A:88:92:78
```

iii) Display average marks of student with registration number "1234":

```
$ grep '^1234:' studentInformation.txt | cut -d ':' -f 6-8 | awk -F ':' '{print ($1+$2+$3)/3}'
70
```

iv) Display the title row in uppercase. Other lines unchanged:

```
$ { head -1 studentInformation.txt | tr 'a-z' 'A-Z'; tail -n +2 studentInformation.txt; }
REGISTRATIONNO:NAME:DEPARTMENT:BRANCH:SECTION:SUB1:SUB2:SUB3
1234:XYZ:ICT:CCE:A:80:60:70
2345:ABC:ICT:IT:B:90:75:85
3456:PQR:ECE:ECE:A:70:80:75
4567:LMN:ICT:IT:A:88:92:78
5678:DEF:CSE:CSE:B:85:65:95
6789:GHI:ICT:CCE:B:60:70:80
7890:JKL:ECE:ECE:A:75:85:90
8901:MNO:ICT:IT:B:82:77:88
9012:PST:CSE:CSE:A:90:80:70
0123:QRS:ICT:CCE:B:85:90:80
```

4. List all files containing "MIT" in the current folder and display lines with "MIT" replaced by "Manipal Institute of Technology":

Assuming some files have "MIT" somewhere:

```
$ grep -l 'MIT' *  # Lists files with MIT

$ grep 'MIT' fileContainingMIT.txt | sed 's/MIT/Manipal Institute of Technology/g'
```

Example:

```
$ echo -e "Welcome to MIT.\nMIT is great institution." > sample.txt

$ grep 'MIT' sample.txt | sed 's/MIT/Manipal Institute of Technology/g'
Welcome to Manipal Institute of Technology.
Manipal Institute of Technology is great institution.
```

5. Display the number of lines, characters, and words of files containing a digit in their name:

```
$ ls *[0-9]*    # lists files with digits in filename

$ wc *[0-9]*    # word count stats for those files
```

6. Run wc command in background many times and kill all wc processes:

Run several background processes:

```
$ wc fruits.txt &
$ wc studentInformation.txt &
$ wc data.txt &

$ jobs   # to check background jobs running
```

Kill all wc processes:

```
$ killall wc
```

OR, if killall not available:

```
$ pkill wc
```

Or find process IDs and kill:

```
$ pgrep wc  # gets process ids

$ kill $(pgrep wc)
```

Additional Exercise Examples

- **Delete the character before the last character in each line:**

```
sed -e 's/.\(.\)$/\1/' filename
```

- **Count number of lines containing digits in a file:**

```
grep -c '[0-9]' filename
```