July-2024

Q-5 (A) Write a command for any six. [6]

1. **Display last line of file f1.**

Tail -1 f1

Sed –n '$p' data

1. **Replace all occurrences of 'he' with 'she' and 'his' with 'her' in file f1.**

sed –e 's/he/she/' –e 's/his/her/' f1

1. **Display only the count of duplicate words in file f1.**

Sort f1 | uniq –dc (d to print only duplicate words, c to count)

1. **Convert file content of f1 to upper case.**

tr '[:lower:]' '[:upper:]' < f1

awk '{ print toupper($0) }' input.txt

1. **Display files of current directory that contains 'unix' in it.**

Grep –l "unix" ./* (-l to list file names only)

1. **Insert blank lines after each line in file f1 except last line.**

Sed 'a\

>

' f1

1. **Display length of longest line of file f1.**

wc –L f1 (-L for max line length)

1. **Convert decimal number 1234 into hexadecimal.**

echo "ibase=10;obase=16;1234" | bc

(B) Write commands using AWK utility. (Any three) [6]

1. Display longest word in each line of file f1.

{

l='';

for(i=1; i<=NF; i++)

{

if(length($i) > length(l))

l=$i;

}

print l

}

2. Display number of words in each line of file f1 without using NF.

{

count=0;

for (i = 1; i <= length($0); i++)

{

if (substr($0, i, 1) ~ /[[:space:]]/)

count++;

}

print count+1;

}

3. Display lines of file f1 which consists of only alphabets.

awk '/^[a-zA-Z]*$/' filename.txt

4. Display lines 5 to 10 of file f1 that do not contain 'unix'.

awk 'NR >= 5 && NR <= 10 && !/unix/' f1

5. Display each line of file f1 in reverse.

```
{
line = $0;
l= length(line);
rev = "";
for (i=l;i>0;i--)
{
rev=rev substr(line, i, 1);
}
print rev;
}
```

March-2024

Q-5 (A) Answer following using grep or sed utility (any six). [6]

1. **Display lines of file f1 that begin with 'The'.**

Grep "^The" f1

Sed –n '/^The/p' f1

1. **Count total number of blank lines of file f1.**

Grep –c "^$" f1

sed -n '/^$/p' filename.txt | wc -l

1. **Display the lines of file f1 that do not contain "Unix".**

Grep –v "Unix" f1

Sed '/Unix/d' f1

1. **Replace all occurrences of "unix" with "linux" of file f1.**

Sed 's/unix/linux/' f1

1. **Display the lines of file f1 that contain "VB.net" and/or "Asp.net".**

Grep –e "VB.net" –e "Asp.net" f1

1. **Display lines of file f1 which have exactly 4 chars.**

grep "^[a-Z]\{4\}$" f1

grep "^. . . . $" f1

sed -n '/^....$/p' f1

1. **Does not display line number 3 to 5 of file f1.**

Sed '3,5d' f1

1. **Insert blank line after each line of file f1.**

Sed 'a\

' f1

(B) Write commands using AWK utility. (Any three) [6]

1. Display those words of file f1 whose length are greater than 5 chars and consist of only alphabets.

```
{
for (i = 1; i <= NF; i++) {
if (length($i) > 5 && $i ~ /^[a-zA-Z]+$/) {
print $i
}
}
}
```

2. Display even numbers of words in each line of file f1.

Awk 'NF%2==0' f1

3. Count occurrences of pattern “unix” in file f1.

```
{
for (i = 1; i <= NF; i++) {
if ($i == “unix”) {
c++;
}
}
} END {print c}
```

4. Count number of vowels in file f1.

```
{
for (i = 1; i <= length($0); i++) {
char = tolower(substr($0, i, 1))
if (char ~ /[aeiou]/) {
count++
}
}
}
END {
print count
}
```

5. Display all palindrome words in file f1.

```
{
for (i = 1; i <= NF; i++) {
word = $i
rev = ""
for (j = length(word); j > 0; j--) {
rev = rev substr(word, j, 1)
}
if (word == rev) {
print word
}
}
}
```

November-2023

Q-4 (A) Write unix commands for following (Any 7) [7]

1. **To count number of words from line 10 to 20 of file test.txt.**

```
Sed –n ’10,20p’ test.txt | wc –w
```

1. **Display the lines which are not starting with 2 at the beginning.**

```
Grep –v “^2” test.txt
Sed ‘/^2/d’ test.txt
```

1. **Display lines of file f1 having 1st and last same characters.**

```
awk 'substr($0, 1, 1) == substr($0, length($0), 1)' filename.txt
```

1. **Remove repeated lines from file test.txt**

```
sort test.txt | uniq
```

1. **Append dashed line after each line of file test.txt**

```
sed ‘a\
---------------------------------
```

‘ test.txt

1. **Replace all occurrences of “SYBCA” with “TYBCA” on lines 5 to 10 of file f1.**

sed ‘5,10s/SYBCA/TYBCA/’ f1

1. **To delete all special characters from file test.txt**

sed 's/[^a-zA-Z0-9]//g' test.txt

1. **To run a utility Pr1 at 11:00 am**

at 11:00 am Pr1

1. **Display file names from current directory whose name start with an alphabet and ends with digit.**

ls | grep '^[a-zA-Z].*[0-9]$'

Q-5 (B) Write commands using AWK utility. (Any four) [4]

1. **Print even numbers of words in each line of file test.txt**

```
{
for (i = 1; i <= NF; i++) {
if (i % 2 == 0) {
printf "%s ", $i
}
}
print ""
}
```

1. **Display words whose length is greater than 6 characters and consists of only alphabets.**

```
{
for (i = 1; i <= NF; i++) {
if (length($i) > 6 && $i ~ /^[a-zA-Z]+$/) {
print $i
}
}
}
```

1. **Print lines 6 to 12 from file text.txt**

awk 'NR >= 6 && NR <= 12' text.txt

1. **Count all occurrences of pattern “TYBCA” in file test.txt**

awk '/nana/ {count++} END{print count}' f1 (1st occurrence only)

OR

```
{
for (i = 1; i <= NF; i++) {
if ($i == “TYBCA”)
count++;
}
} END {print count}
```

1. **Display lines of file f1 in uppercase**

awk '{ print toupper($0) }' f1

1. **Display lines of file f1 in reverse**

```
{
line = $0
rev = ""
for (i = length(line); i > 0; i--) {
rev = rev substr(line, i, 1)
}
```

```
print rev

}
```

Excel Data

{"group":"limit","unwind":"redact "}
{"group":"sort","unwind":"bucket"}
{"group":"sum","unwind":"geonear"}
{"group":"distinct"}
{"group":"count"}
{"group":"match"}
{"group":"project"}
{"group":"push"}
{"group":"all"}
{"group":"unset"}
{"group":"skip"}
{"group":"facet"}
{"group":"fill"}