Assignment-6

Find and awk

1) By using find command search all empty files in a given dir and also delete those files.

a) If we want to search empty files from current working directory and display then

```
bhushan@ubuntu:~/findprac$ ls
emptydir   emptydir5  file2.txt  file4.txt  file6.txt
emptydir3  file1.txt  file3.txt  file5.txt  file7.txt
bhushan@ubuntu:~/findprac$ find . -type f -empty
./file4.txt
./file3.txt
./file1.txt
./file2.txt
./file7.txt
./file6.txt
bhushan@ubuntu:~/findprac$
```

b) If we want to search empty directories from current working directory and display then

```
bhushan@ubuntu:~/findprac$ find . -type d -empty
./emptydir3
./emptydir5
./emptydir
bhushan@ubuntu:~/findprac$
```

c) If we want to search empty files from current working directory and delete it then

```
bhushan@ubuntu:~/findprac$ ls
emptydir   emptydir5  file2.txt  file4.txt  file6.txt
emptydir3  file1.txt  file3.txt  file5.txt  file7.txt
bhushan@ubuntu:~/findprac$ find . -type f -empty -delete
bhushan@ubuntu:~/findprac$ ls
emptydir  emptydir3  emptydir5  file5.txt
bhushan@ubuntu:~/findprac$
```

Awk

Point 1: Default behavior of awk command

```
bhushan@ubuntu:~/Hardsoft$ cat file4.txt
   ID          Name        Salary      Country
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
  [103]       -Loki        55000       Germany
  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk '{print}' file4.txt
   ID          Name        Salary      Country
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
  [103]       -Loki        55000       Germany
  [104]       -Hina        35000        India
```

Point 2: If we want to display with specified pattern then

```
bhushan@ubuntu:~/Hardsoft$ awk '{print}' file4.txt
   ID          Name        Salary      Country
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
  [103]       -Loki        55000       Germany
  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk '/Ind/ {print}' file4.txt
  [101]       -Rutu        -25000       -India
  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk '/00/ {print}' file4.txt
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
  [103]       -Loki        55000       Germany
  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk '/t/ {print}' file4.txt
   ID          Name        Salary      Country
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
```

Point 3: If we specify the column number on this command, it will print that line only.

```
bhushan@ubuntu:~/Hardsoft$ cat file4.txt
   ID          Name        Salary      Country
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
  [103]       -Loki        55000       Germany
  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk '{print $1 $3}' file4.txt
IDSalary
[101]-25000
[102]45000
[103]55000
[104]35000
bhushan@ubuntu:~/Hardsoft$ awk '{print $1 $4}' file4.txt
IDCountry
[101]-India
[102]Belgium
[103]Germany
[104]India
```

Point 4: To display the line number in output, use the NR variable with the Awk command
NR: It is used to show the current count of the lines. The awk command performs action once for each line. These lines are said as records.

```
bhushan@ubuntu:~/Hardsoft$ awk '{print NR $0}' file4.txt
1   ID          Name        Salary      Country
2  [101]       -Rutu        -25000       -India
3  [102]       -Bont        45000       Belgium
4  [103]       -Loki        55000       Germany
5  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk '{print NR $2}' file4.txt
1Name
2-Rutu
3-Bont
4-Loki
5-Hina
```

Point 5: To display the last field of the file, execute the NF variable with the Awk command
NF: It is used to count the number of fields within the current database.

```
bhushan@ubuntu:~/Hardsoft$ awk '{print NF}' file4.txt
4
4
4
4
4
bhushan@ubuntu:~/Hardsoft$ awk '{print $NF}' file4.txt
Country
-India
Belgium
Germany
India
```

Point 6: To separate the output by a '-' symbol or (:) semicolon, specify it with ORS command
ORS: It is used to store the output record separator. It separates the output records. It prints the content of the ORS command automatically.

```
bhushan@ubuntu:~/Hardsoft$ awk 'BEGIN { ORS ="-"} {print $1}' file4.txt
ID-[101]-[102]-[103]-[104]-bhushan@ubuntu:~/Hardsoft$
bhushan@ubuntu:~/Hardsoft$ awk 'BEGIN { ORS ="-"} {print $3}' file4.txt
Salary--25000-45000-55000-35000-bhushan@ubuntu:~/Hardsoft$ awk 'BEGIN { ORS ="-"
} {print $0}' file4.txt
   ID          Name        Salary      Country -  [101]       -Rutu        -2500
0       -India -  [102]       -Bont        45000       Belgium-  [103]       -Lo
ki        55000       Germany-  [104]       -Hina        35000        India-bhus
han@ubuntu:~/Hardsoft$
```

Point 7: To print the numbers from 1 to 8

```
han@ubuntu:~/Hardsoft$ awk 'BEGIN { for(i=1;i<=8;i++) print "square of", i, "is"
,i*i; }'
square of 1 is 1
square of 2 is 4
square of 3 is 9
square of 4 is 16
square of 5 is 25
square of 6 is 36
square of 7 is 49
square of 8 is 64
```

Point 8: To calculate the third column of the created data

```
bhushan@ubuntu:~/Hardsoft$ cat file4.txt
   ID          Name        Salary      Country
  [101]       -Rutu        -25000       -India
  [102]       -Bont        45000       Belgium
  [103]       -Loki        55000       Germany
  [104]       -Hina        35000        India
bhushan@ubuntu:~/Hardsoft$ awk -F" " '{x+=$3}END{print x}' file4.txt
110000
```

Point 9 : To find the value of exp 8

```
bhushan@ubuntu:~/Hardsoft$ awk 'BEGIN{x=exp(8); print x}'
2980.96
```