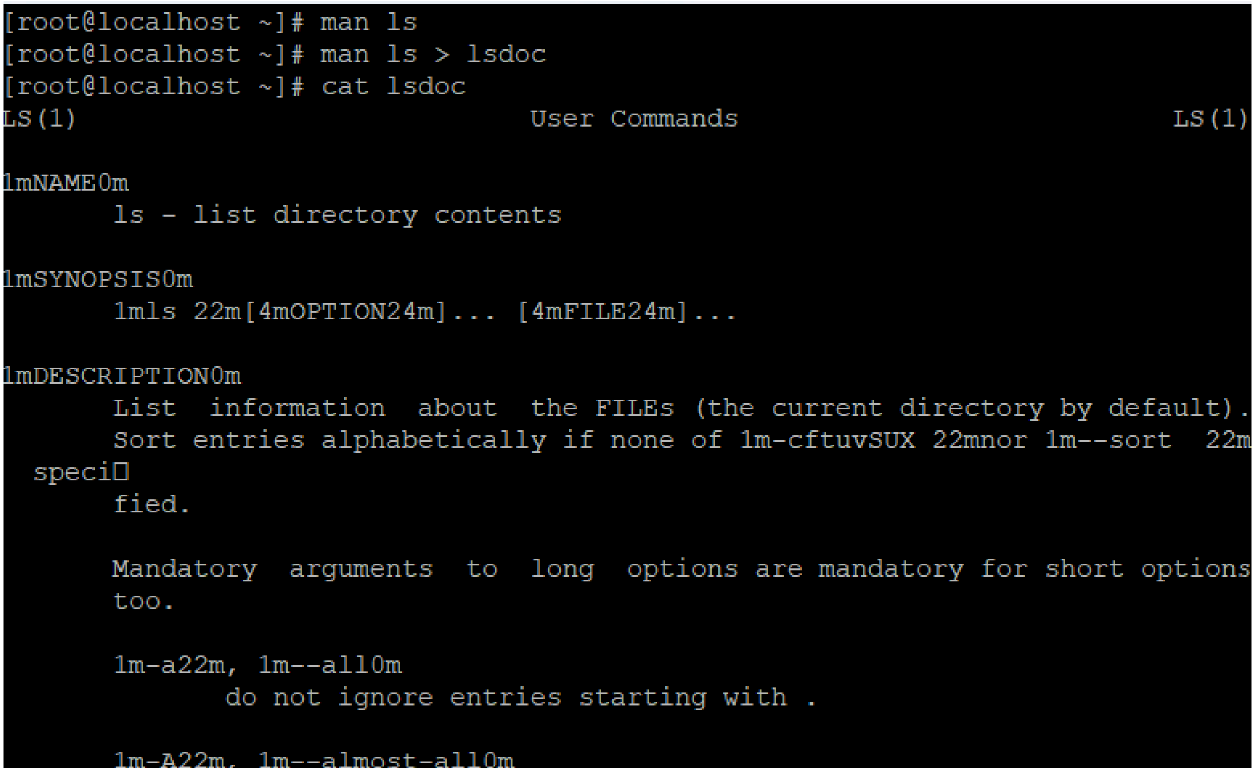
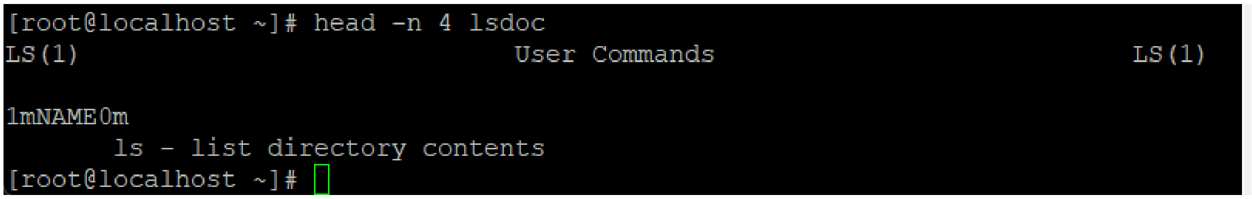
1: Redirect the content of the help document (man) ls, into a file called as lsdoc.



```
[root@localhost ~]# man ls
[root@localhost ~]# man ls > lsdoc
[root@localhost ~]# cat lsdoc
LS(1)                            User Commands                           LS(1)

1mNAME0m
       ls - list directory contents

1mSYNOPSIS0m
       1mls 22m[4mOPTION24m]... [4mFILE24m]...

1mDESCRIPTION0m
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of 1m-cftuvSUX 22mnor 1m--sort  22m
  speci□
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       1m-a22m, 1m--all0m
              do not ignore entries starting with .

       1m-A22m, 1m--almost-all0m
```

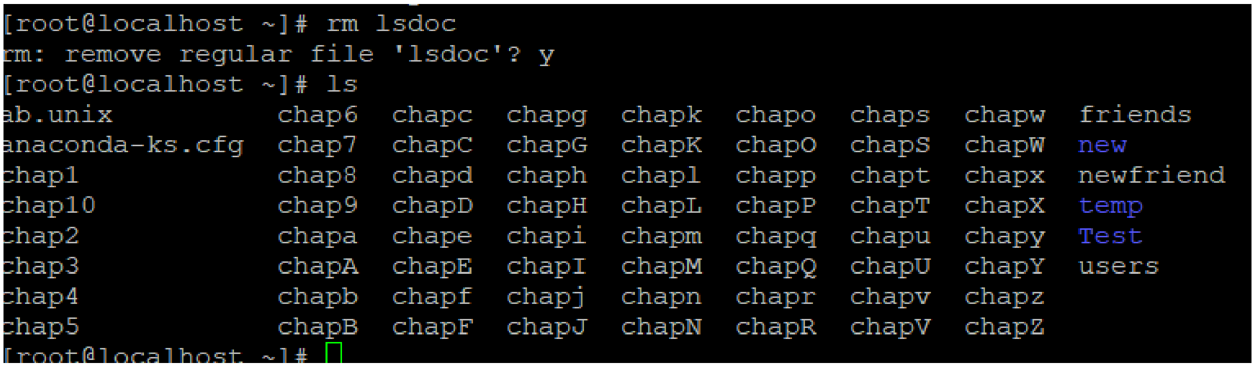
2: Display the content of the lsdoc page wise.
>> less lsdoc
Press space to move to the next page , b for back page and q for quit

3: Display only the first 4 lines of the lsdoc file.



```
[root@localhost ~]# head -n 4 lsdoc
LS(1)                            User Commands                           LS(1)

1mNAME0m
       ls - list directory contents
[root@localhost ~]# 
```

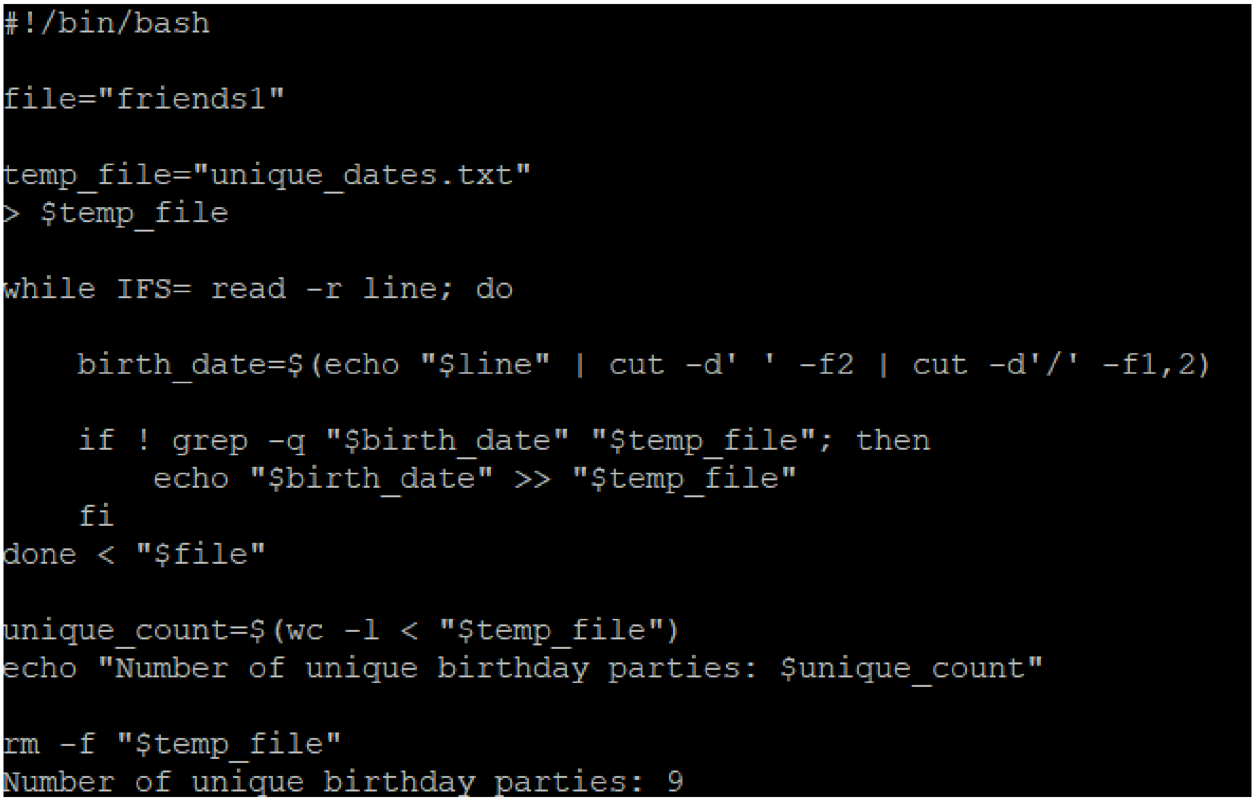
5: Remove the file lsdoc.



```
[root@localhost ~]# rm lsdoc
rm: remove regular file 'lsdoc'? y
[root@localhost ~]# ls
ab.unix          chap6  chapc  chapg  chapk  chapo  chaps  chapw  friends
anaconda-ks.cfg  chap7  chapC  chapG  chapK  chapO  chapS  chapW  new
chap1            chap8  chapd  chaph  chapl  chapp  chapt  chapx  newfriend
chap10           chap9  chapD  chapH  chapL  chapP  chapT  chapX  temp
chap2            chapa  chape  chapi  chapm  chapq  chapu  chapy  Test
chap3            chapA  chapE  chapI  chapM  chapQ  chapU  chapY  users
chap4            chapb  chapf  chapj  chapn  chapr  chapv  chapz
chap5            chapB  chapF  chapJ  chapN  chapR  chapV  chapZ
[root@localhost ~]# 
```

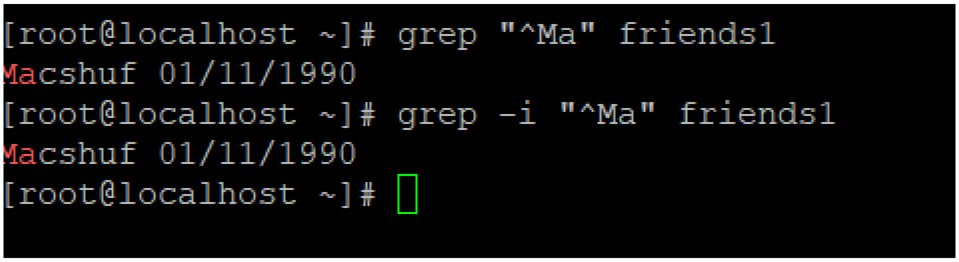
6: There will be B’day celebration from the friends file, find how many B’day parties will be held. If two of the friends have the B’date on the same day, then we will be having one party on that day.

```
[hpcsa@localhost ~]$ cat friends
Shakir 01/10/1993
Mohan 02/10/1996
Hamid 22/12/1993
Macshuf 01/11/1990
Deepak 31/01/1993
Ganesh 09/01/1997
Sarthak 10/01/1994
Tanzeem 08/08/1992
Abhishek 10/01/1994
Palash 06/07/1994
Swapnil 01/10/1993
Vidhi 08/08/1994
Swapnali 08/08/1994
```



```
#!/bin/bash

file="friends1"

temp_file="unique_dates.txt"
> $temp_file

while IFS= read -r line; do

    birth_date=$(echo "$line" | cut -d' ' -f2 | cut -d'/' -f1,2)

    if ! grep -q "$birth_date" "$temp_file"; then
        echo "$birth_date" >> "$temp_file"
    fi
done < "$file"

unique_count=$(wc -l < "$temp_file")
echo "Number of unique birthday parties: $unique_count"

rm -f "$temp_file"
Number of unique birthday parties: 9
```

7: Display the lines starting with Ma, in the file friends

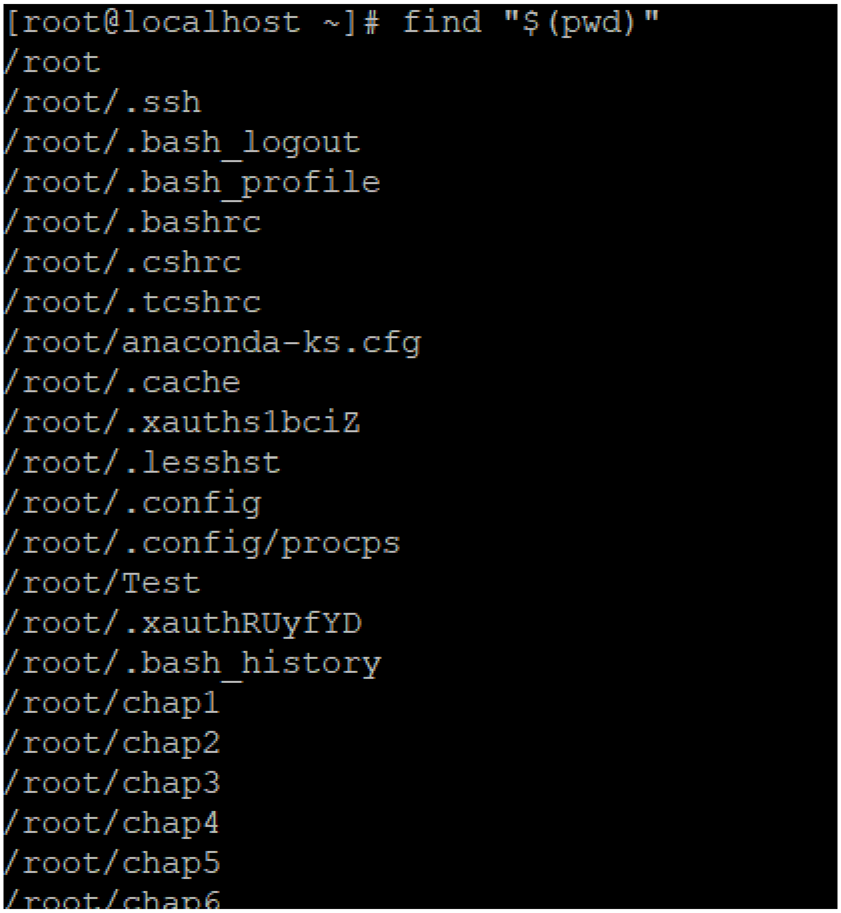


```
[root@localhost ~]# grep "^Ma" friends1
Macshuf 01/11/1990
[root@localhost ~]# grep -i "^Ma" friends1
Macshuf 01/11/1990
[root@localhost ~]# 
```

.

8: Display the lines starting with Ma, ending with i or ending with id, in the file friends.
>>grep -E "^Ma.*(i|id)$" friends1

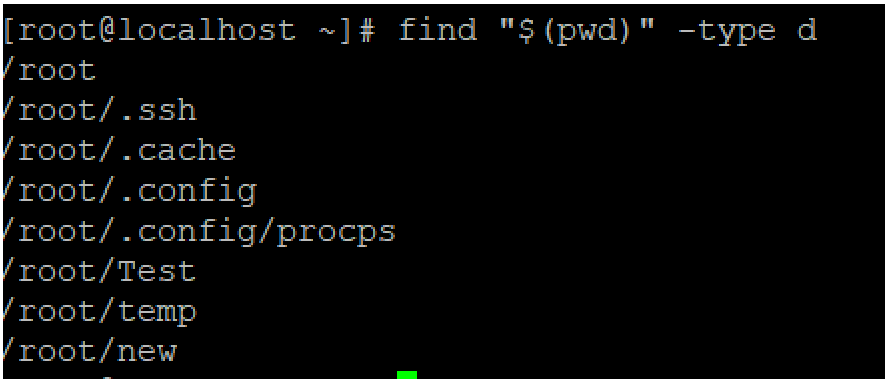
9: Print all the files and the directory files from the current directory across all the sub directories, along with its path



```
[root@localhost ~]# find "$(pwd)"
/root
/root/.ssh
/root/.bash_logout
/root/.bash_profile
/root/.bashrc
/root/.cshrc
/root/.tcshrc
/root/anaconda-ks.cfg
/root/.cache
/root/.xauths1bciZ
/root/.lesshst
/root/.config
/root/.config/procps
/root/Test
/root/.xauthRUyfYD
/root/.bash_history
/root/chap1
/root/chap2
/root/chap3
/root/chap4
/root/chap5
/root/chap6
```

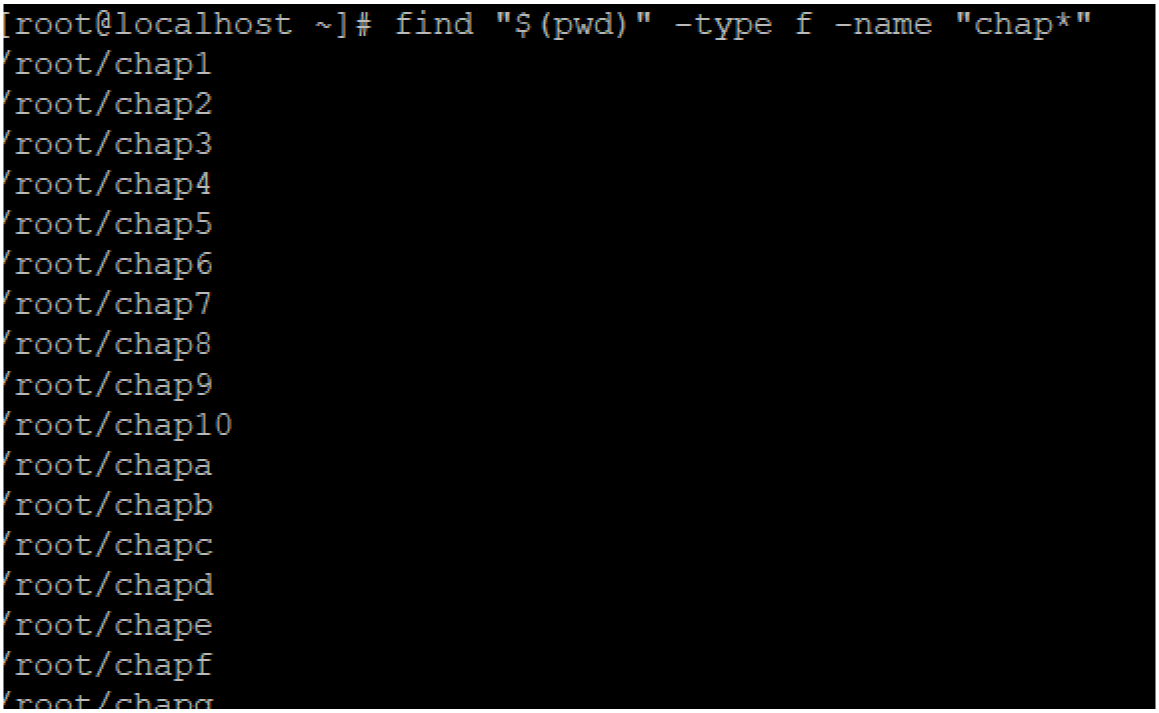
.

10: Print only the Directory files.



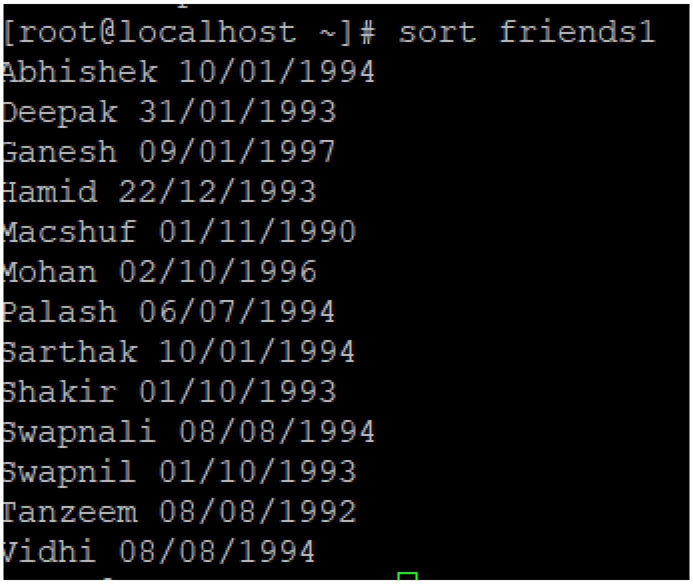
```
[root@localhost ~]# find "$(pwd)" -type d
/root
/root/.ssh
/root/.cache
/root/.config
/root/.config/procps
/root/Test
/root/temp
/root/new
```

11: Display the files starting with chap, along with its path.



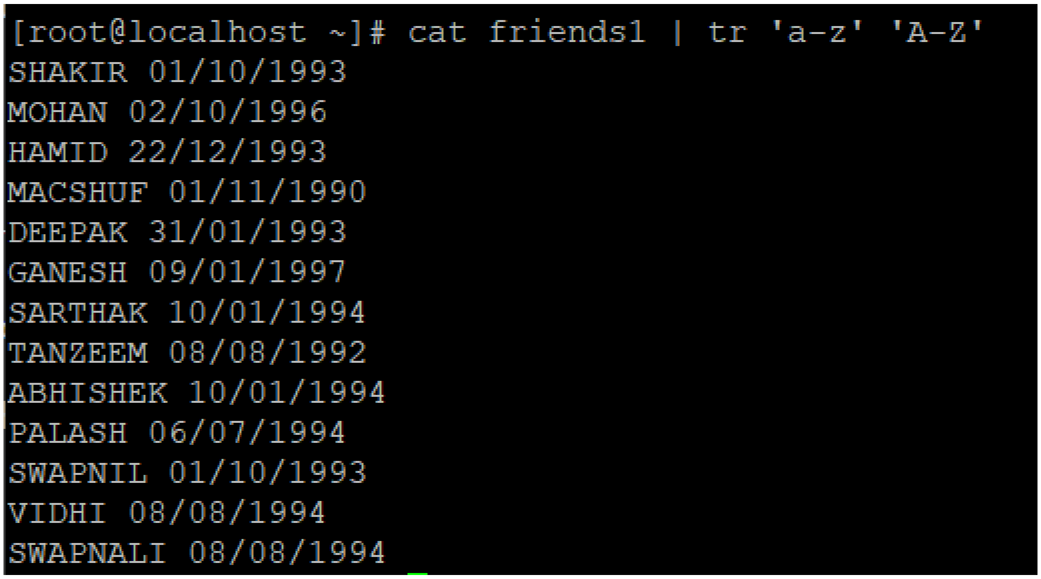
```
[root@localhost ~]# find "$(pwd)" -type f -name "chap*"
/root/chap1
/root/chap2
/root/chap3
/root/chap4
/root/chap5
/root/chap6
/root/chap7
/root/chap8
/root/chap9
/root/chap10
/root/chapa
/root/chapb
/root/chapc
/root/chapd
/root/chape
/root/chapf
/root/chapg
```

12. Sort the file friends in ascending order of names.



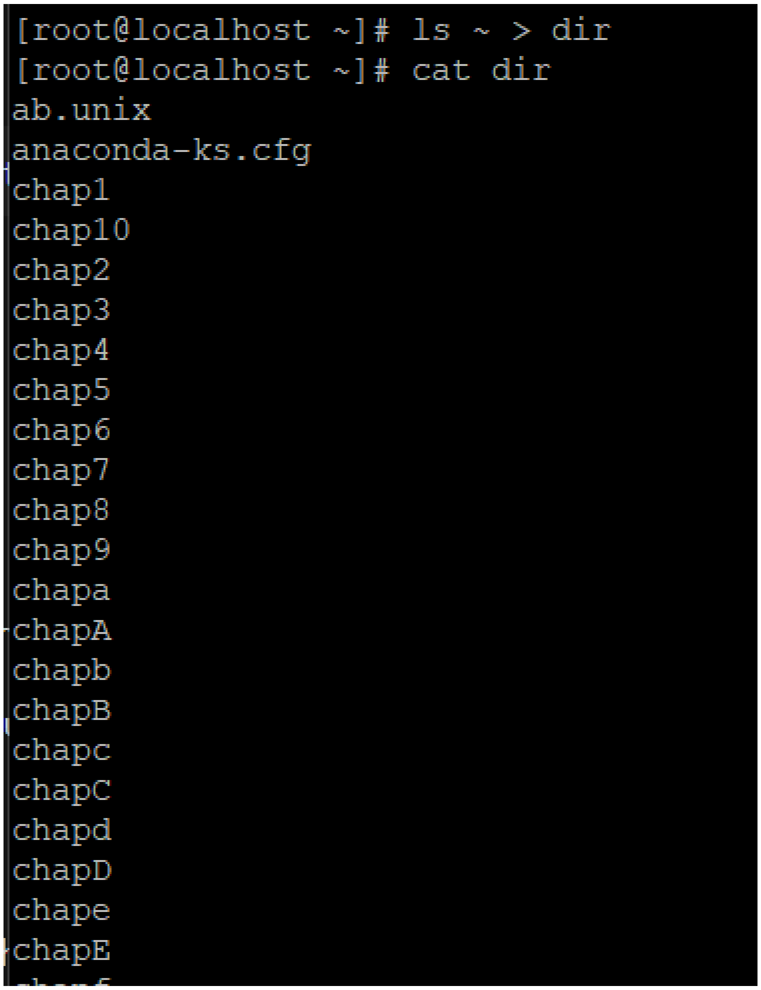
```
[root@localhost ~]# sort friends1
Abhishek 10/01/1994
Deepak 31/01/1993
Ganesh 09/01/1997
Hamid 22/12/1993
Macshuf 01/11/1990
Mohan 02/10/1996
Palash 06/07/1994
Sarthak 10/01/1994
Shakir 01/10/1993
Swapnali 08/08/1994
Swapnil 01/10/1993
Tanzeem 08/08/1992
Vidhi 08/08/1994
```

13: Display the contents of the file friends in uppercase letters.



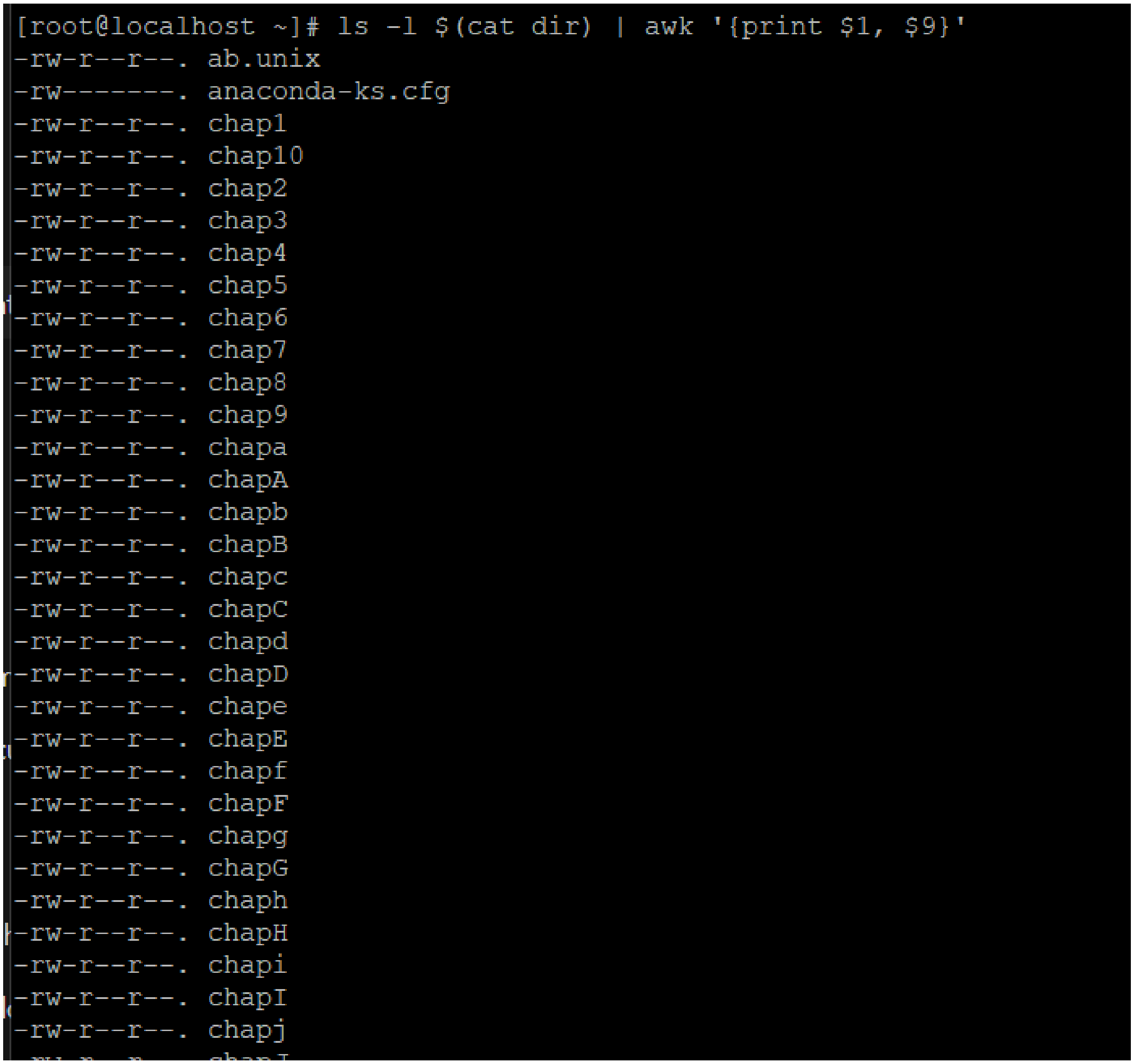
```
[root@localhost ~]# cat friends1 | tr 'a-z' 'A-Z'
SHAKIR 01/10/1993
MOHAN 02/10/1996
HAMID 22/12/1993
MACSHUF 01/11/1990
DEEPAK 31/01/1993
GANESH 09/01/1997
SARTHAK 10/01/1994
TANZEEM 08/08/1992
ABHISHEK 10/01/1994
PALASH 06/07/1994
SWAPNIL 01/10/1993
VIDHI 08/08/1994
SWAPNALI 08/08/1994
```

14: Store the contents of your home directory in a file called dir.



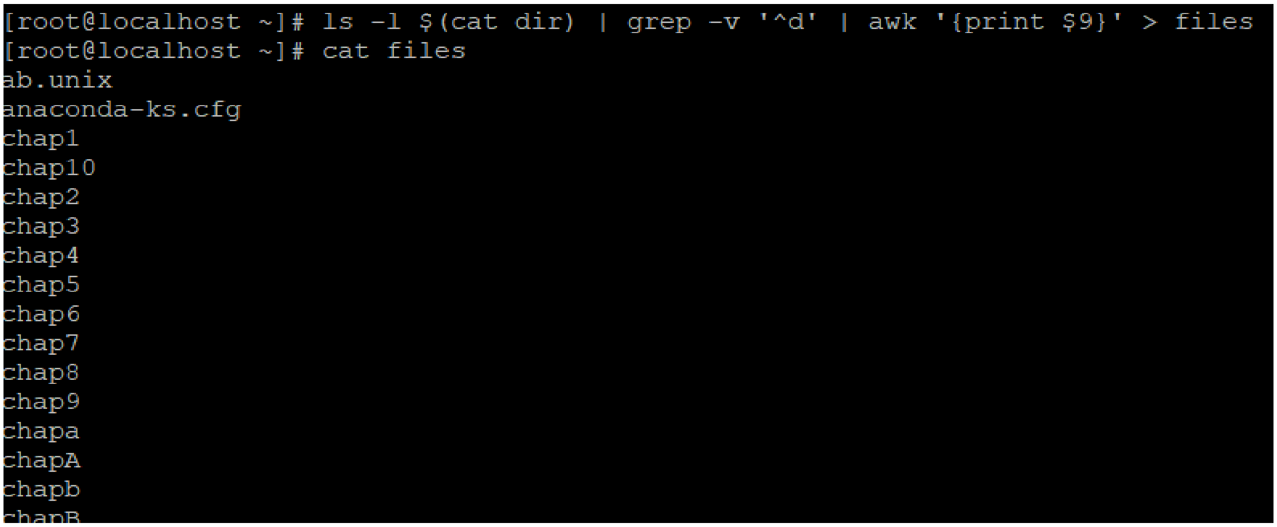
```
[root@localhost ~]# ls ~ > dir
[root@localhost ~]# cat dir
ab.unix
anaconda-ks.cfg
chap1
chap10
chap2
chap3
chap4
chap5
chap6
chap7
chap8
chap9
chapa
chapA
chapb
chapB
chapc
chapC
chapd
chapD
chape
chapE
chapf
```

15: From the above file dir, display the file permissions and the name of the file only.



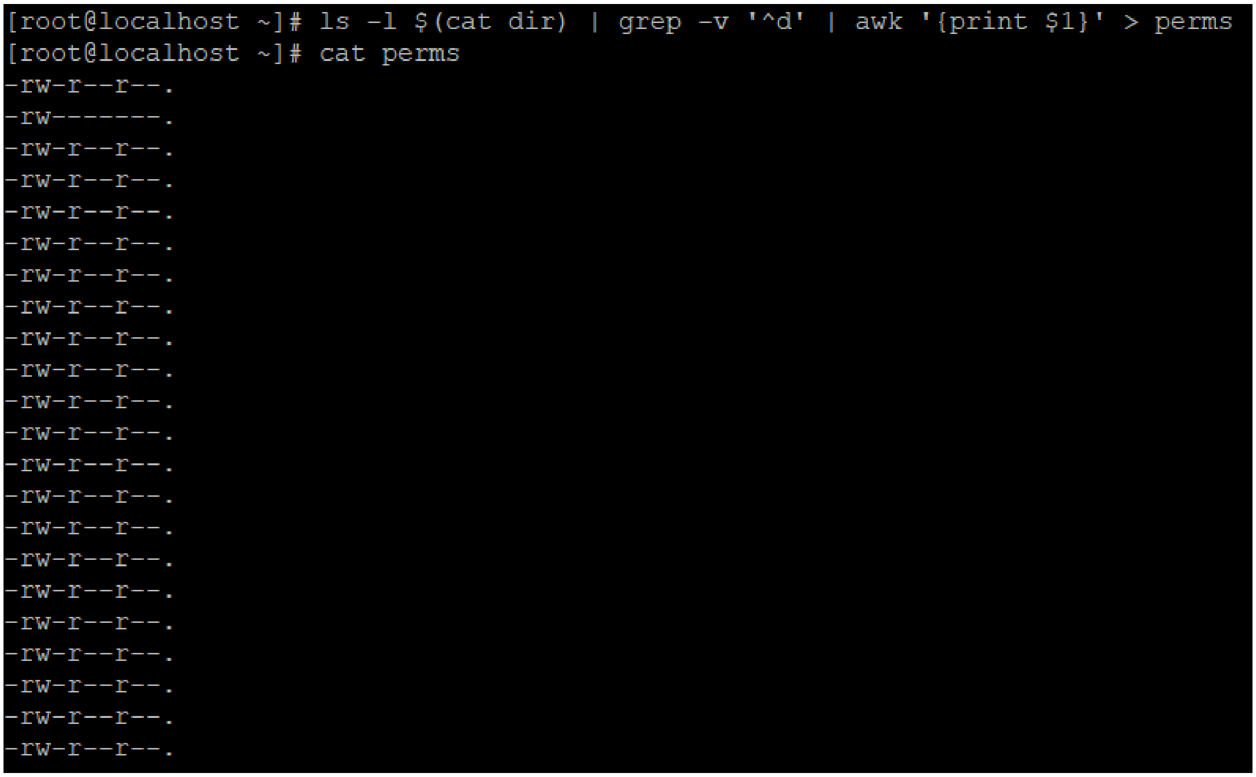
```
[root@localhost ~]# ls -l $(cat dir) | awk '{print $1, $9}'
-rw-r--r--. ab.unix
-rw-------. anaconda-ks.cfg
-rw-r--r--. chap1
-rw-r--r--. chap10
-rw-r--r--. chap2
-rw-r--r--. chap3
-rw-r--r--. chap4
-rw-r--r--. chap5
-rw-r--r--. chap6
-rw-r--r--. chap7
-rw-r--r--. chap8
-rw-r--r--. chap9
-rw-r--r--. chapa
-rw-r--r--. chapA
-rw-r--r--. chapb
-rw-r--r--. chapB
-rw-r--r--. chapc
-rw-r--r--. chapC
-rw-r--r--. chapd
-rw-r--r--. chapD
-rw-r--r--. chape
-rw-r--r--. chapE
-rw-r--r--. chapf
-rw-r--r--. chapF
-rw-r--r--. chapg
-rw-r--r--. chapG
-rw-r--r--. chaph
-rw-r--r--. chapH
-rw-r--r--. chapi
-rw-r--r--. chapI
-rw-r--r--. chapj
```

16: From the same dir file, store only the file names in a file called files.



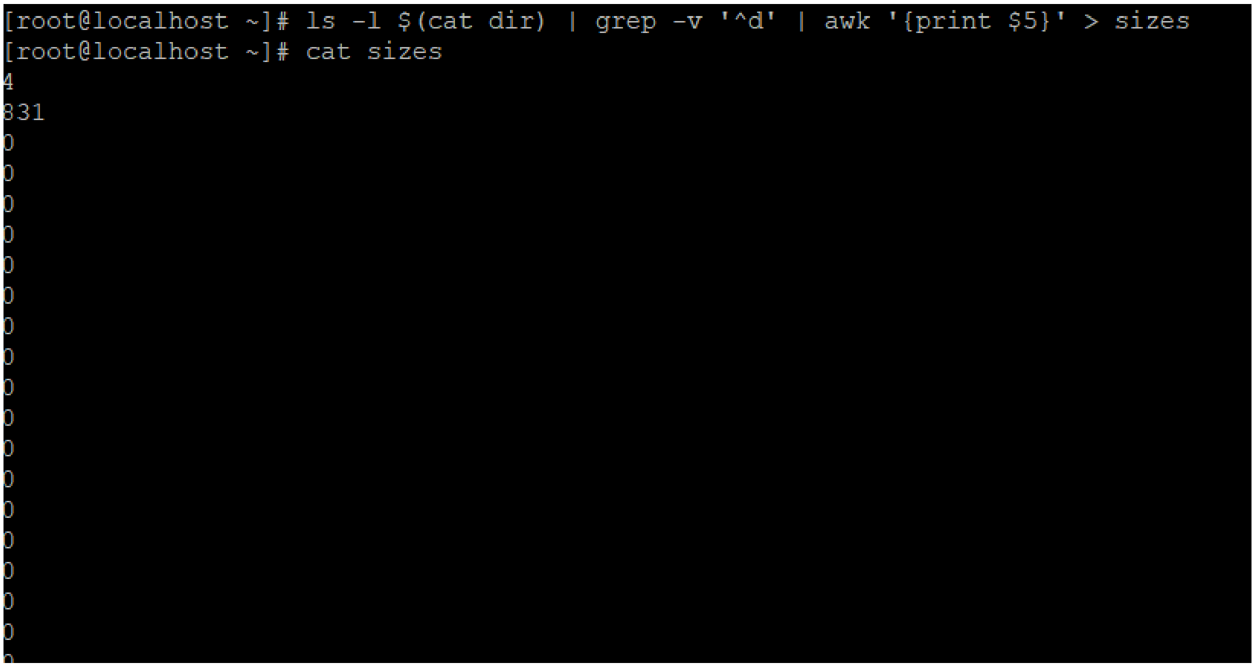
```
[root@localhost ~]# ls -l $(cat dir) | grep -v '^d' | awk '{print $9}' > files
[root@localhost ~]# cat files
ab.unix
anaconda-ks.cfg
chap1
chap10
chap2
chap3
chap4
chap5
chap6
chap7
chap8
chap9
chapa
chapA
chapb
chapB
```

17: From the same dir file, store only the permissions of files in a file called perms.



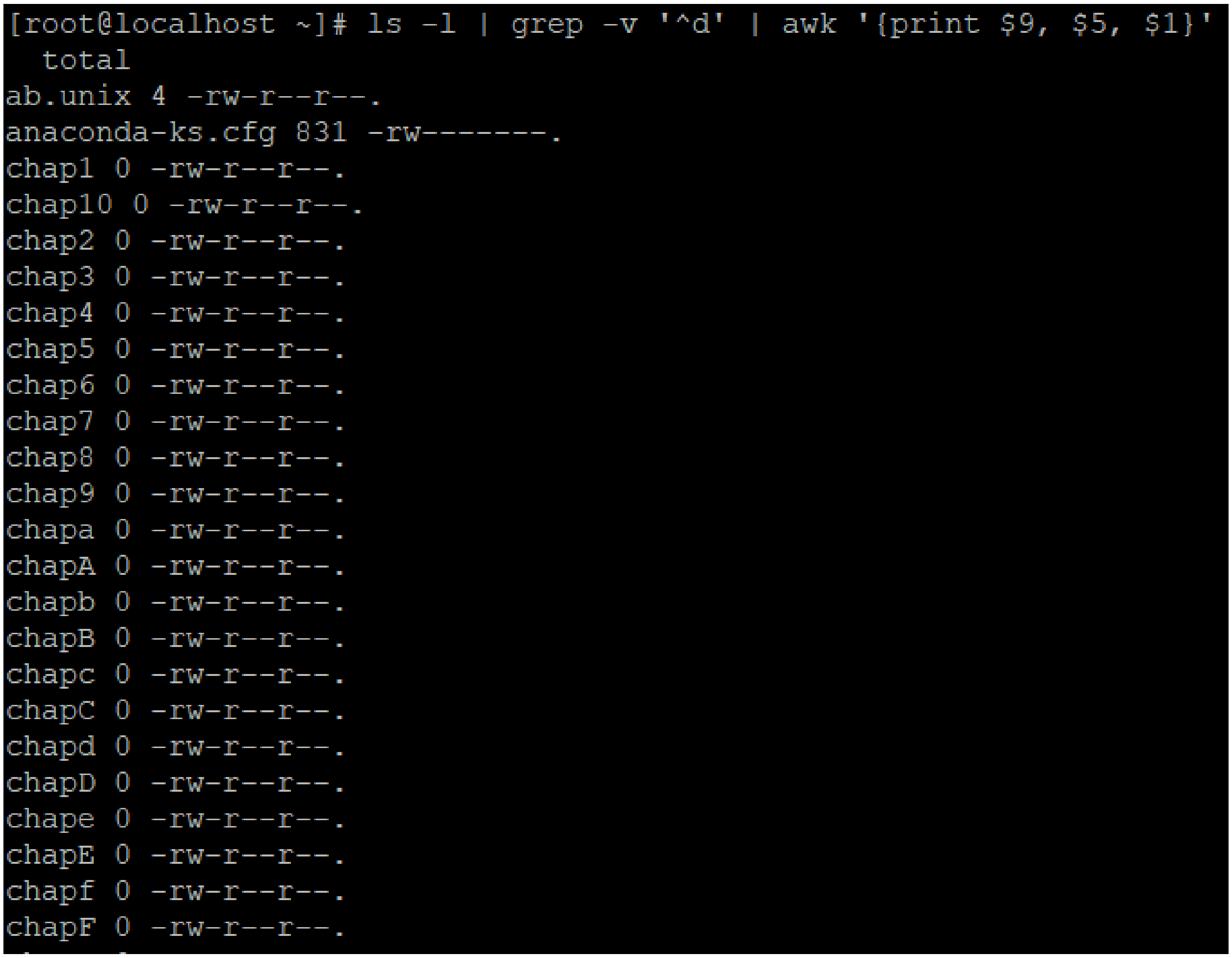
```
[root@localhost ~]# ls -l $(cat dir) | grep -v '^d' | awk '{print $1}' > perms
[root@localhost ~]# cat perms
-rw-r--r--.
-rw-------.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
-rw-r--r--.
```

18: From the same dir file, store only the file sizes in a file called sizes.



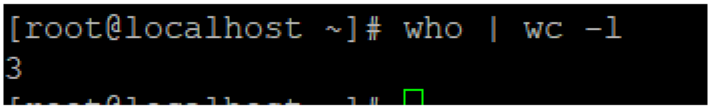
```
[root@localhost ~]# ls -l $(cat dir) | grep -v '^d' | awk '{print $5}' > sizes
[root@localhost ~]# cat sizes
4
831
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
```

19: Display the file names, sizes and permissions from your directory in that order.



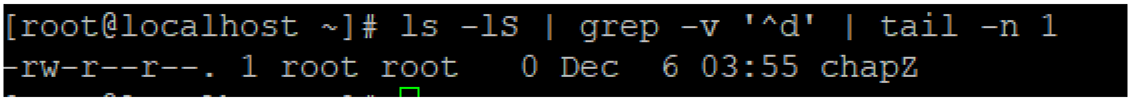
```
[root@localhost ~]# ls -l | grep -v '^d' | awk '{print $9, $5, $1}'
 total
ab.unix 4 -rw-r--r--.
anaconda-ks.cfg 831 -rw-------.
chap1 0 -rw-r--r--.
chap10 0 -rw-r--r--.
chap2 0 -rw-r--r--.
chap3 0 -rw-r--r--.
chap4 0 -rw-r--r--.
chap5 0 -rw-r--r--.
chap6 0 -rw-r--r--.
chap7 0 -rw-r--r--.
chap8 0 -rw-r--r--.
chap9 0 -rw-r--r--.
chapa 0 -rw-r--r--.
chapA 0 -rw-r--r--.
chapb 0 -rw-r--r--.
chapB 0 -rw-r--r--.
chapc 0 -rw-r--r--.
chapC 0 -rw-r--r--.
chapd 0 -rw-r--r--.
chapD 0 -rw-r--r--.
chape 0 -rw-r--r--.
chapE 0 -rw-r--r--.
chapf 0 -rw-r--r--.
chapF 0 -rw-r--r--.
```

20: Display the number of users working on the system



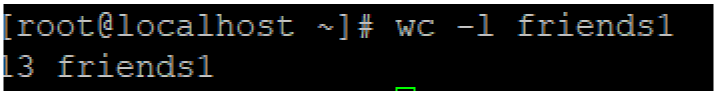
```
[root@localhost ~]# who | wc -l
3
```

21: Find out the smallest file in your directory.



```
[root@localhost ~]# ls -lS | grep -v '^d' | tail -n 1
-rw-r--r--. 1 root root    0 Dec  6 03:55 chapZ
```

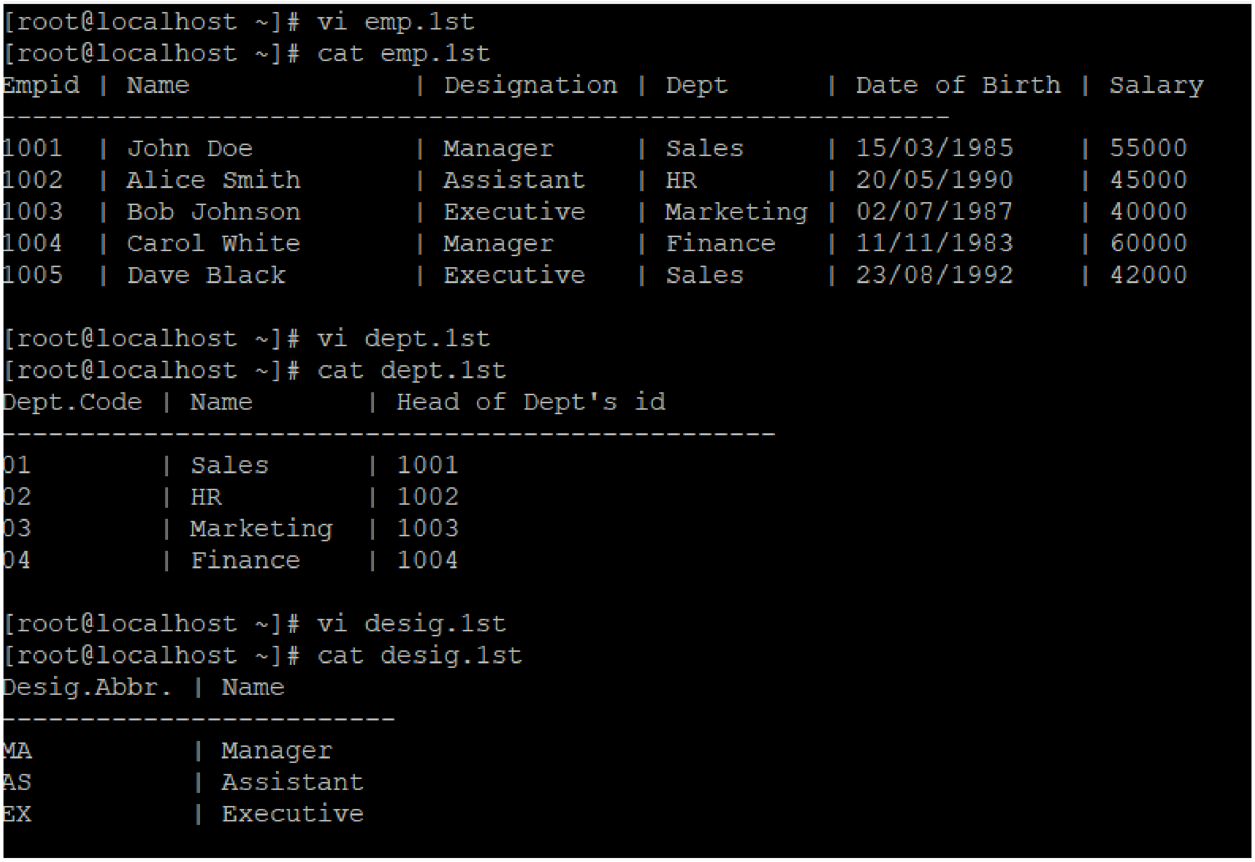
22: Display the total number of lines present in the file friends.



```
[root@localhost ~]# wc -l friends1
13 friends1
```

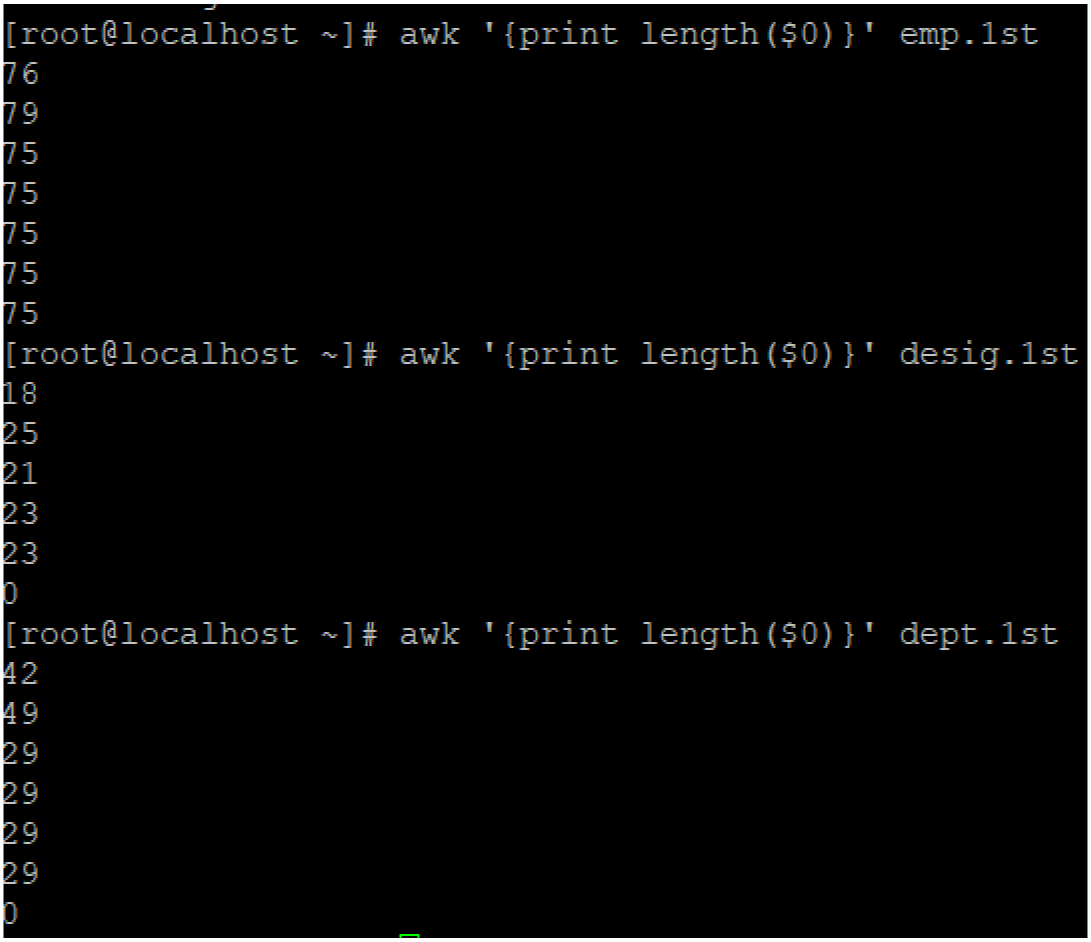
23:Create the following fixed record format files (with “|” delimiter between fields) with the structure given below, and populate them with relevant data use these files to solve following questions.

emp.lst: Empid(4),Name(18),Designation(9),Dept(10),Date of Birth(8),Salary(5)
dept.lst : Dept.Code(2),Name(10),Head of Dept’s id(4)
desig.lst: Designation Abbr.(2), Name (9)



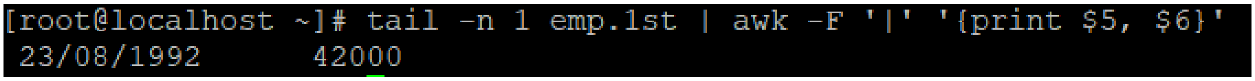
```
[root@localhost ~]# vi emp.1st
[root@localhost ~]# cat emp.1st
Empid | Name              | Designation | Dept      | Date of Birth | Salary
------------------------------------------------------------
1001  | John Doe          | Manager     | Sales     | 15/03/1985    | 55000
1002  | Alice Smith       | Assistant   | HR        | 20/05/1990    | 45000
1003  | Bob Johnson       | Executive   | Marketing | 02/07/1987    | 40000
1004  | Carol White       | Manager     | Finance   | 11/11/1983    | 60000
1005  | Dave Black        | Executive   | Sales     | 23/08/1992    | 42000

[root@localhost ~]# vi dept.1st
[root@localhost ~]# cat dept.1st
Dept.Code | Name       | Head of Dept's id
-------------------------------------------------
01        | Sales      | 1001
02        | HR         | 1002
03        | Marketing  | 1003
04        | Finance    | 1004

[root@localhost ~]# vi desig.1st
[root@localhost ~]# cat desig.1st
Desig.Abbr. | Name
-------------------------
MA          | Manager
AS          | Assistant
EX          | Executive
```

1. Find the record lengths of each file.



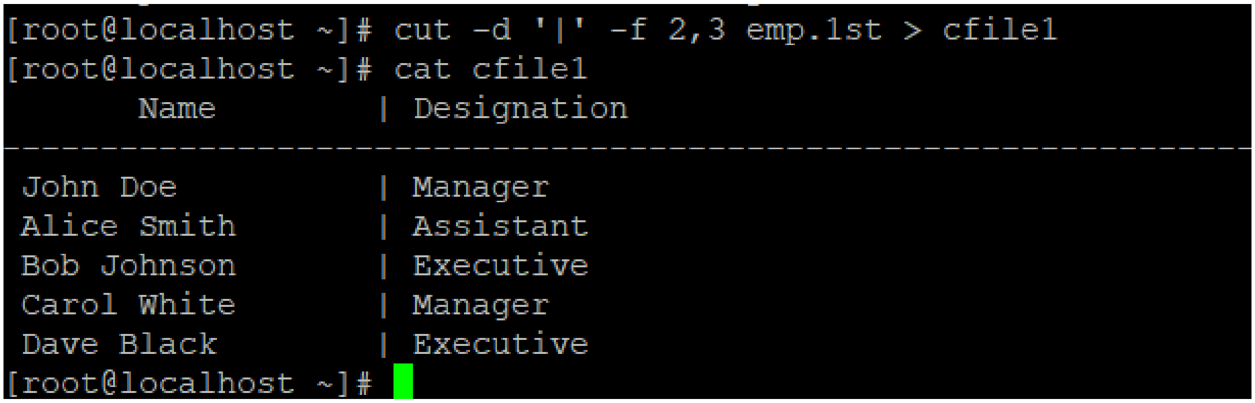
```
[root@localhost ~]# awk '{print length($0)}' emp.1st
76
79
75
75
75
75
75
[root@localhost ~]# awk '{print length($0)}' desig.1st
18
25
21
23
23
0
[root@localhost ~]# awk '{print length($0)}' dept.1st
42
49
29
29
29
29
0
```

2. Display only the date of birth and salary of the last employee record.



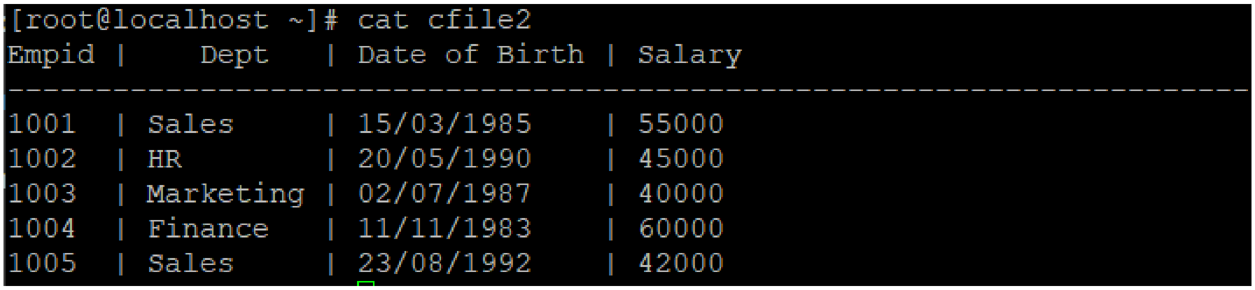
```
[root@localhost ~]# tail -n 1 emp.1st | awk -F '|' '{print $5, $6}'
 23/08/1992      42000
```

3. Extract only employee names and designations. (Use column specifications). Save output as cfile1.



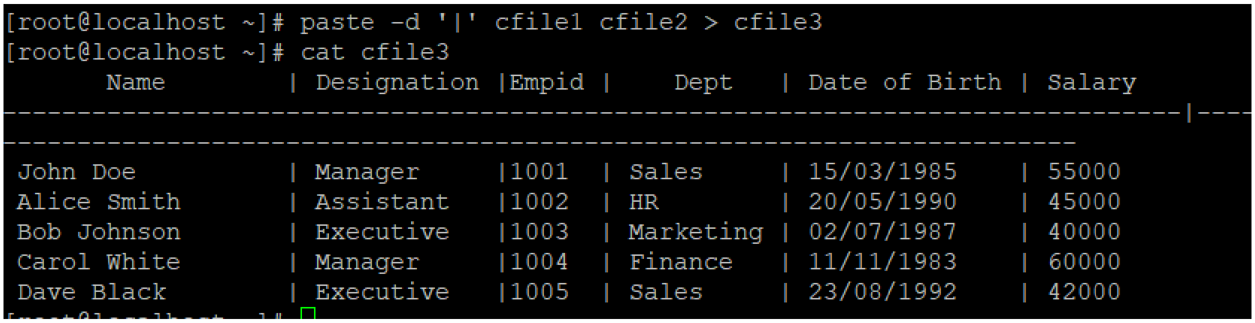
```
[root@localhost ~]# cut -d '|' -f 2,3 emp.1st > cfile1
[root@localhost ~]# cat cfile1
       Name        | Designation
----------------------------------------------------------------
 John Doe          | Manager
 Alice Smith       | Assistant
 Bob Johnson       | Executive
 Carol White       | Manager
 Dave Black        | Executive
[root@localhost ~]#
```

4. Extract Emp.id, dept, dob and salary. (Use field specifications). Save output as cfile2.



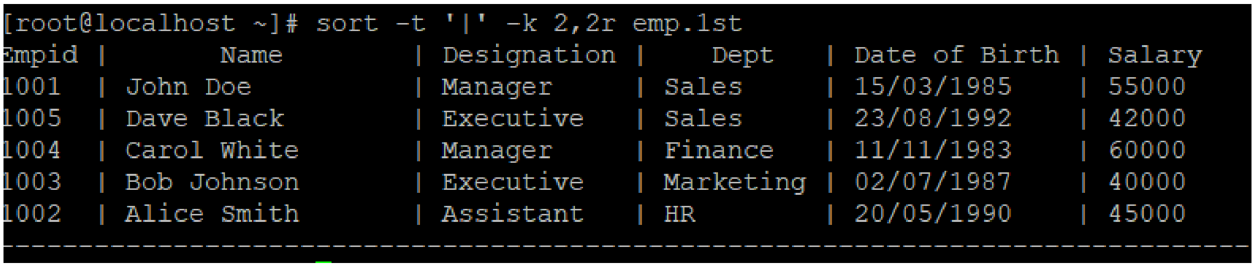
```
[root@localhost ~]# cat cfile2
Empid |    Dept    | Date of Birth | Salary
-----------------------------------------------------------------------
1001  | Sales      | 15/03/1985    | 55000
1002  | HR         | 20/05/1990    | 45000
1003  | Marketing  | 02/07/1987    | 40000
1004  | Finance    | 11/11/1983    | 60000
1005  | Sales      | 23/08/1992    | 42000
```

5. Fix the files cfile1 and cfile2 laterally, along with the delimiter.



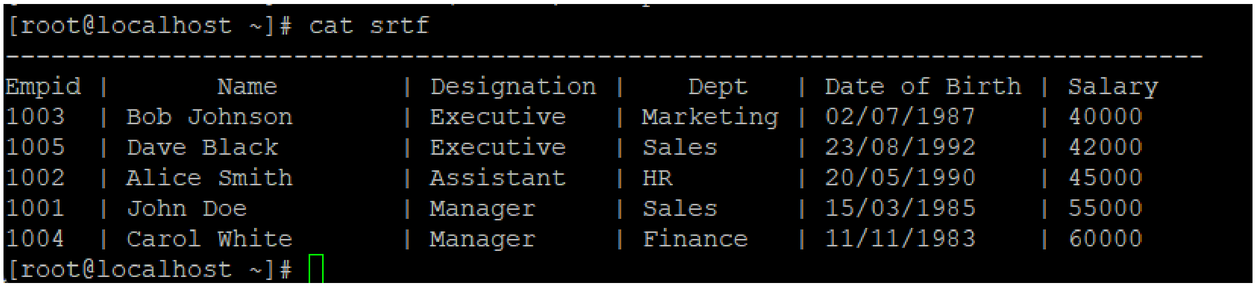
```
[root@localhost ~]# paste -d '|' cfile1 cfile2 > cfile3
[root@localhost ~]# cat cfile3
       Name        | Designation |Empid |    Dept   | Date of Birth | Salary
-------------------------------------------------------------------------------|----
------------------------------------------------------------------------
 John Doe          | Manager     |1001  | Sales     | 15/03/1985    | 55000
 Alice Smith       | Assistant   |1002  | HR        | 20/05/1990    | 45000
 Bob Johnson       | Executive   |1003  | Marketing | 02/07/1987    | 40000
 Carol White       | Manager     |1004  | Finance   | 11/11/1983    | 60000
 Dave Black        | Executive   |1005  | Sales     | 23/08/1992    | 42000
```

6. Sort the emp.lst file in reverse order of Emp. Names.



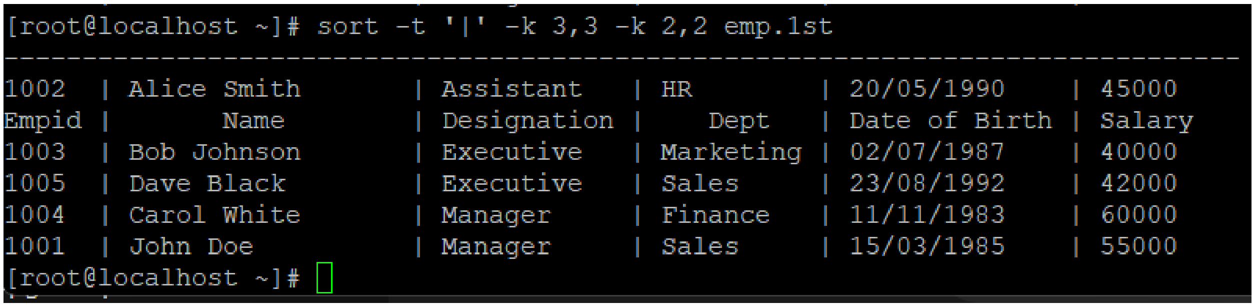
```
[root@localhost ~]# sort -t '|' -k 2,2r emp.1st
Empid |       Name        | Designation |    Dept   | Date of Birth | Salary
1001  | John Doe          | Manager     | Sales     | 15/03/1985    | 55000
1005  | Dave Black        | Executive   | Sales     | 23/08/1992    | 42000
1004  | Carol White       | Manager     | Finance   | 11/11/1983    | 60000
1003  | Bob Johnson       | Executive   | Marketing | 02/07/1987    | 40000
1002  | Alice Smith       | Assistant   | HR        | 20/05/1990    | 45000
-------------------------------------------------------------------------------
```

7. Sort the emp.lst file on the salary field, and store the result in file srtf.



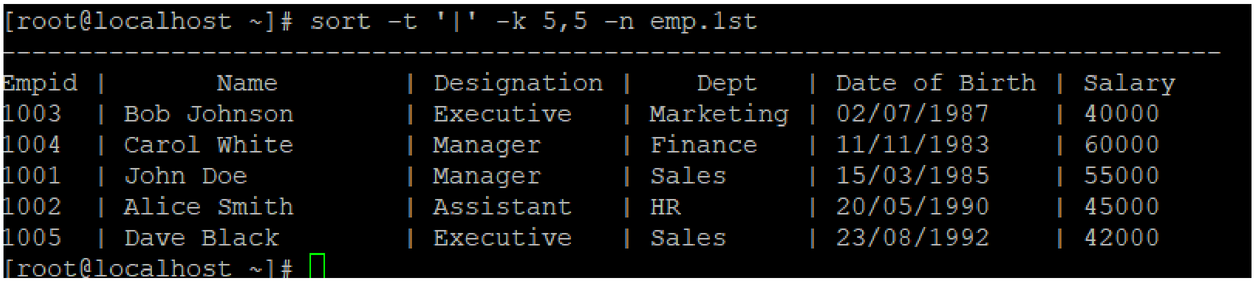
```
[root@localhost ~]# cat srtf
-------------------------------------------------------------------------------
Empid |       Name        | Designation |    Dept   | Date of Birth | Salary
1003  | Bob Johnson       | Executive   | Marketing | 02/07/1987    | 40000
1005  | Dave Black        | Executive   | Sales     | 23/08/1992    | 42000
1002  | Alice Smith       | Assistant   | HR        | 20/05/1990    | 45000
1001  | John Doe          | Manager     | Sales     | 15/03/1985    | 55000
1004  | Carol White       | Manager     | Finance   | 11/11/1983    | 60000
[root@localhost ~]#
```

8. Sort the emp.lst file on designation followed by name.



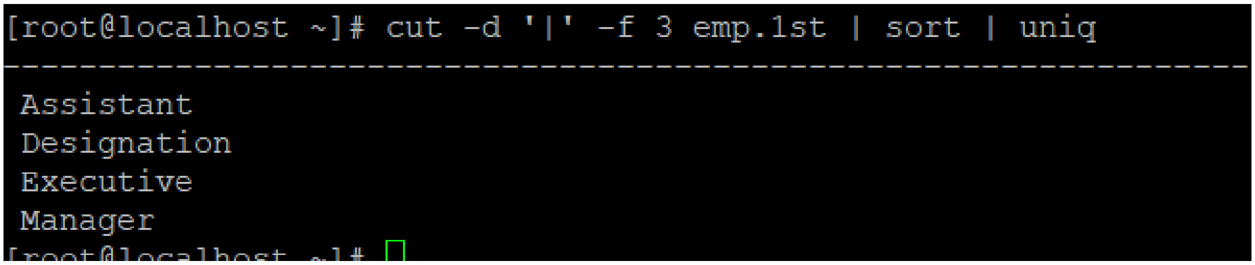
```
[root@localhost ~]# sort -t '|' -k 3,3 -k 2,2 emp.1st
-------------------------------------------------------------------------------
1002  | Alice Smith       | Assistant   | HR        | 20/05/1990    | 45000
Empid |       Name        | Designation |    Dept   | Date of Birth | Salary
1003  | Bob Johnson       | Executive   | Marketing | 02/07/1987    | 40000
1005  | Dave Black        | Executive   | Sales     | 23/08/1992    | 42000
1004  | Carol White       | Manager     | Finance   | 11/11/1983    | 60000
1001  | John Doe          | Manager     | Sales     | 15/03/1985    | 55000
[root@localhost ~]#
```

9. Sort the emp.lst file on the year of birth.



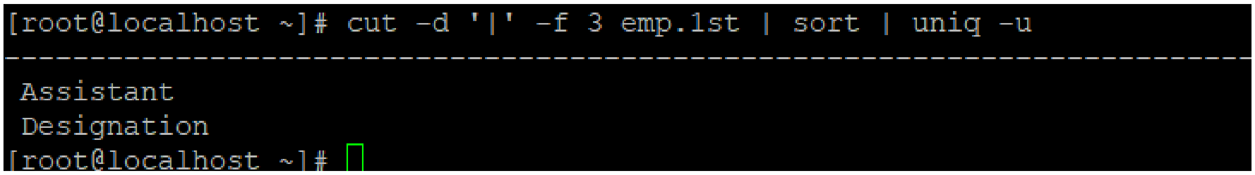
```
[root@localhost ~]# sort -t '|' -k 5,5 -n emp.1st
-------------------------------------------------------------------------------
Empid |        Name         | Designation |    Dept    | Date of Birth | Salary
1003  | Bob Johnson         | Executive   | Marketing  | 02/07/1987    | 40000
1004  | Carol White         | Manager     | Finance    | 11/11/1983    | 60000
1001  | John Doe            | Manager     | Sales      | 15/03/1985    | 55000
1002  | Alice Smith         | Assistant   | HR         | 20/05/1990    | 45000
1005  | Dave Black          | Executive   | Sales      | 23/08/1992    | 42000
[root@localhost ~]#
```

10. Find out the various designations in the employee file. Eliminate duplicate listing of Designations.



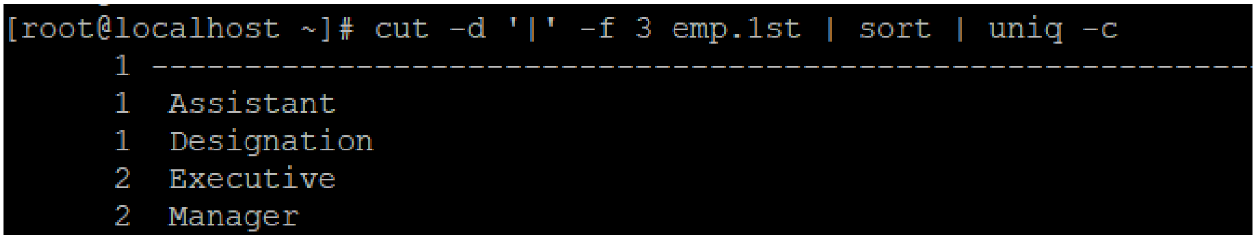
```
[root@localhost ~]# cut -d '|' -f 3 emp.1st | sort | uniq
-----------------------------------------------------------------
 Assistant
 Designation
 Executive
 Manager
[root@localhost ~]#
```

11. Find the non-repeated designation in the employee file.



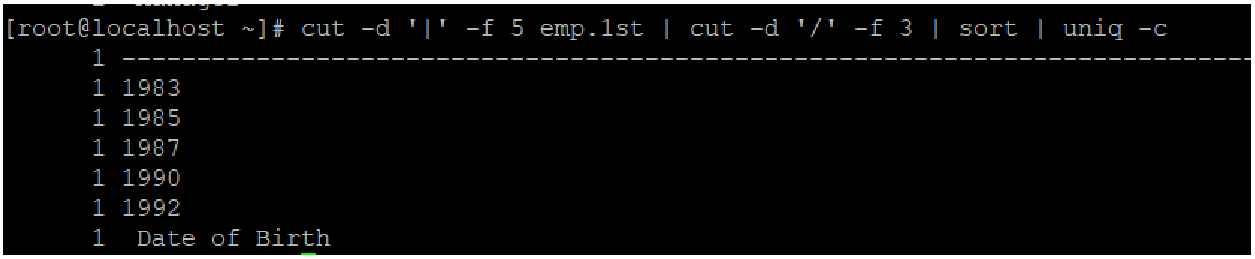
```
[root@localhost ~]# cut -d '|' -f 3 emp.1st | sort | uniq -u
-----------------------------------------------------------------------
 Assistant
 Designation
[root@localhost ~]#
```

12. Find the number of employees with various designations in the employee file.



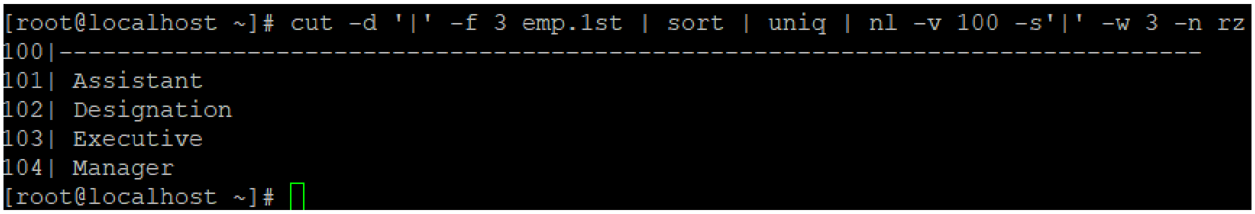
```
[root@localhost ~]# cut -d '|' -f 3 emp.1st | sort | uniq -c
      1 --------------------------------------------------------
      1  Assistant
      1  Designation
      2  Executive
      2  Manager
```

13. Create a listing of the years in which employees were born in, along with number of employees born in that year.



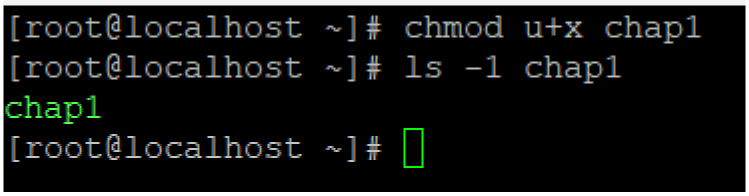
```
[root@localhost ~]# cut -d '|' -f 5 emp.1st | cut -d '/' -f 3 | sort | uniq -c
      1 ---------------------------------------------------------------------------
      1 1983
      1 1985
      1 1987
      1 1990
      1 1992
      1  Date of Birth
```

14. Use nl command to create a code table for designations to include designation code (Start with dept. code 100, and subsequently 105, 110 ...).



```
[root@localhost ~]# cut -d '|' -f 3 emp.1st | sort | uniq | nl -v 100 -s'|' -w 3 -n rz
100|---------------------------------------------------------------------------------
101| Assistant
102| Designation
103| Executive
104| Manager
[root@localhost ~]# 
```

24. Give the execute permission for the user for a file chap1.



```
[root@localhost ~]# chmod u+x chap1
[root@localhost ~]# ls -1 chap1
chap1
[root@localhost ~]# 
```

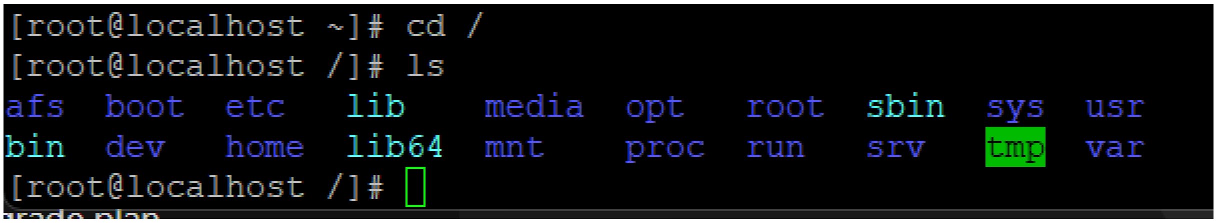
>> ls -l chap1

25. Give the execute permission for user, group and others for a file add.c
>>chmod a+x add.c
>>ls -l add.c

26. Remove the execute permission from user, give read permission to group and others for a file aa.c.
>>chmod u-x,g+r,o+r add.c

27. Give execute permission for users for a.c, kk.c, nato and myfile using single command.
>> chmod u+x a.c kk.c nato myfile

28. Change the directory to root directory. Check the system directories, like bin, etc, usr etc.



```
[root@localhost ~]# cd /
[root@localhost /]# ls
afs  boot  etc   lib    media  opt   root  sbin  sys  usr
bin  dev   home  lib64  mnt    proc  run   srv   tmp  var
[root@localhost /]# 
```