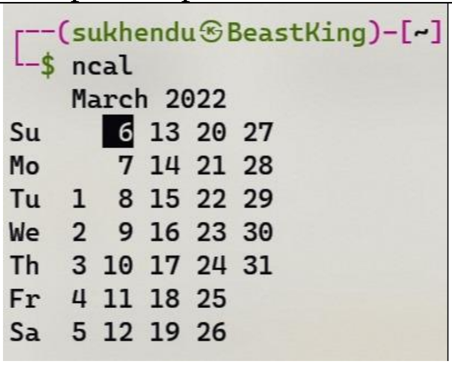
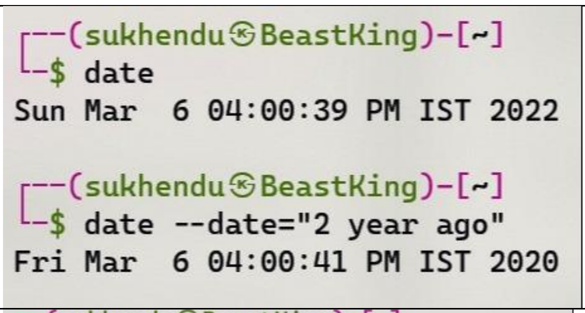
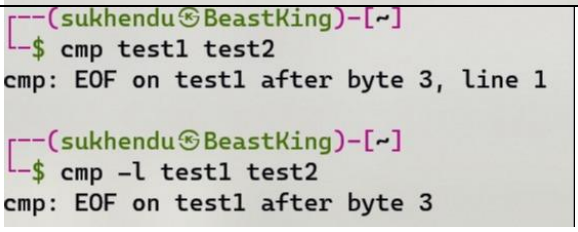
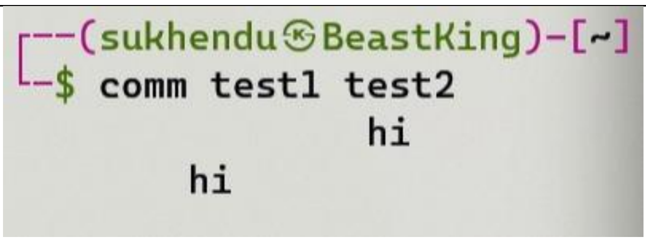
0. <u>For each command, give a brief description of what it does and two examples of how it can be used</u>

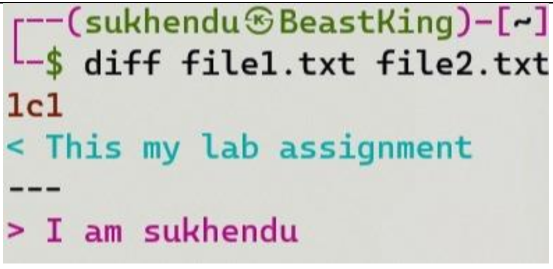
Command	Description	Syntax	Sample Output

cal	**cal** command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.	$ cal [[month] year]	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ ncal`
			`    March 2022`
			`Su     6 13 20 27`
			`Mo     7 14 21 28`
			`Tu  1  8 15 22 29`
			`We  2  9 16 23 30`
			`Th  3 10 17 24 31`
			`Fr  4 11 18 25`
			`Sa  5 12 19 26`

date	Shows the date and time to the nearest second.	$ date [OPTION]... [+FORMAT]	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ date`
			`Sun Mar  6 04:00:39 PM IST 2022`
			
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ date --date="2 year ago"`
			`Fri Mar  6 04:00:41 PM IST 2020`

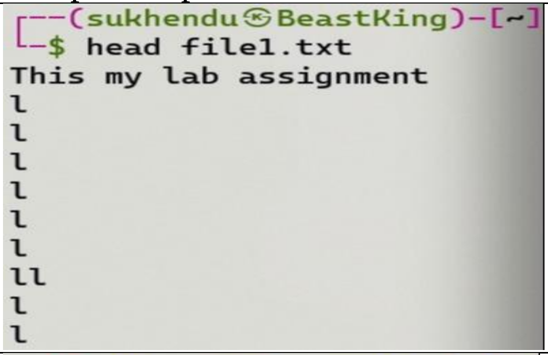
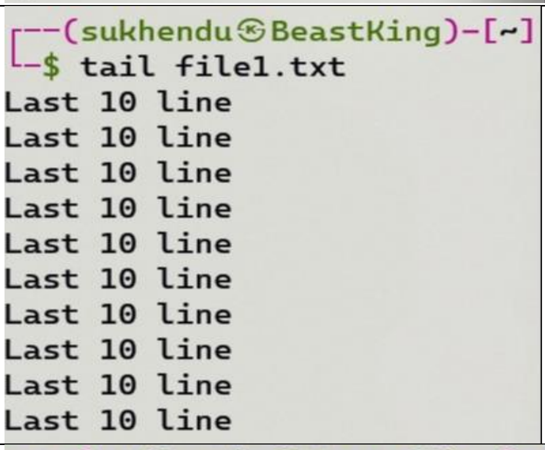
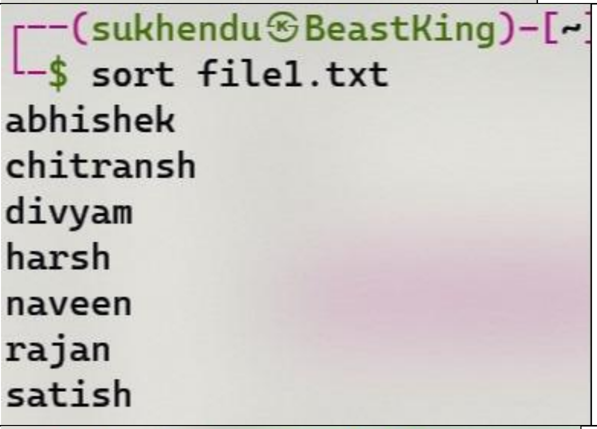
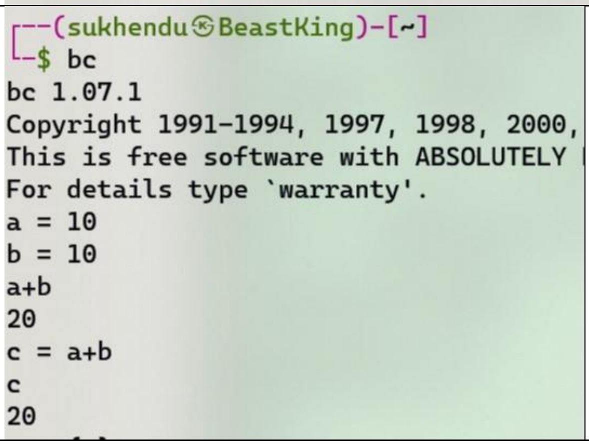
cmp	It compares two files of any type and writes the results to the standard output.	$ cmp [FILE1 OPTION]... [FILE2 [SKIP1 [SKIP2]]]	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ cmp test1 test2`
			`cmp: EOF on test1 after byte 3, line 1`
			
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ cmp -l test1 test2`
			`cmp: EOF on test1 after byte 3`

comm	The comm command compares two sorted files line by line.	$comm file1 file2	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ comm test1 test2`
			`                hi`
			`        hi`

diff	It tells which lines of one file have to be changed to make two files identical.	$diff file1 file2	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ diff file1.txt file2.txt`
			`1c1`
			`< This my lab assignment`
			`---`
			`> I am sukhendu`

Name: Roll No:

Section: Year:

Command	Description	Syntax	Sample Output
head	head by default, prints the first 10 lines of each FILE to standard output.	$head filename	
tail	Print the last 10 lines of each FILE to standard output.	$tail filename	
sort	SORT command is used to sort a file, arranging the records in a particular order.	$sort filename	
bc	**bc** command is used for command line calculator.	$bc	

```
┌──(sukhendu㉿BeastKing)-[~]
└─$ head file1.txt
This my lab assignment
l
l
l
l
l
l
ll
l
l
```

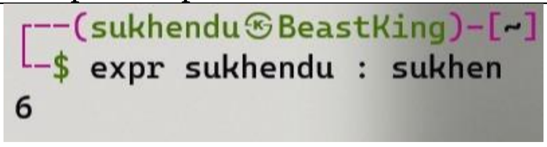
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ tail file1.txt
Last 10 line
Last 10 line
Last 10 line
Last 10 line
Last 10 line
Last 10 line
Last 10 line
Last 10 line
Last 10 line
Last 10 line
```

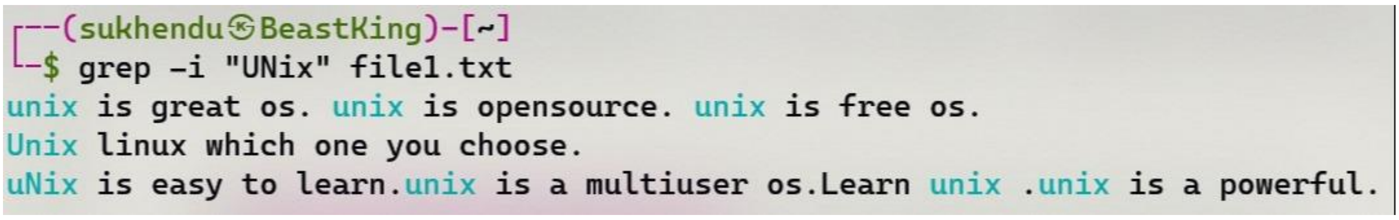
```
┌──(sukhendu㉿BeastKing)-[~
└─$ sort file1.txt
abhishek
chitransh
divyam
harsh
naveen
rajan
satish
```

```
┌──(sukhendu㉿BeastKing)-[~]
└─$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000,
This is free software with ABSOLUTELY
For details type `warranty'.
a = 10
b = 10
a+b
20
c = a+b
c
20
```

Name: Roll No:

Section: Year:

Command	Description	Syntax	Sample Output
expr	The **expr** command in Unix evaluates a given expression and displays its corresponding output.	$expr expression	 ┌──(sukhendu㉿BeastKing)-[~] └─$ expr sukhendu : sukhen 6
grep	The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.	$grep -i ‘unix” filename	



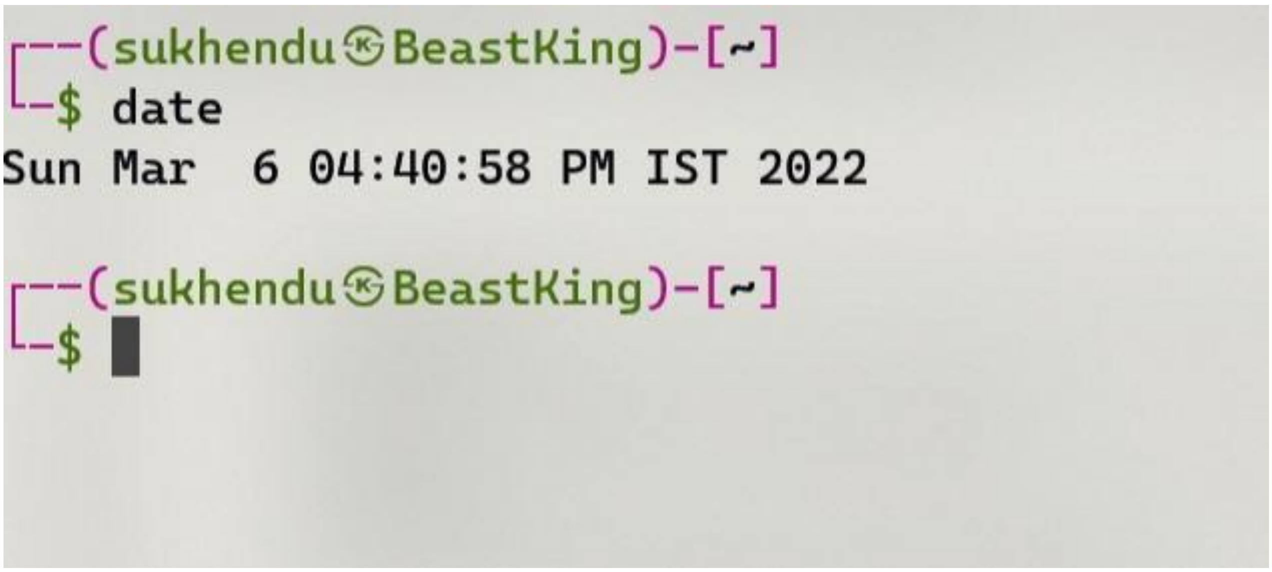
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ grep -i "UNix" file1.txt
unix is great os. unix is opensource. unix is free os.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

1.

a. Display the current time in 12-hour format.

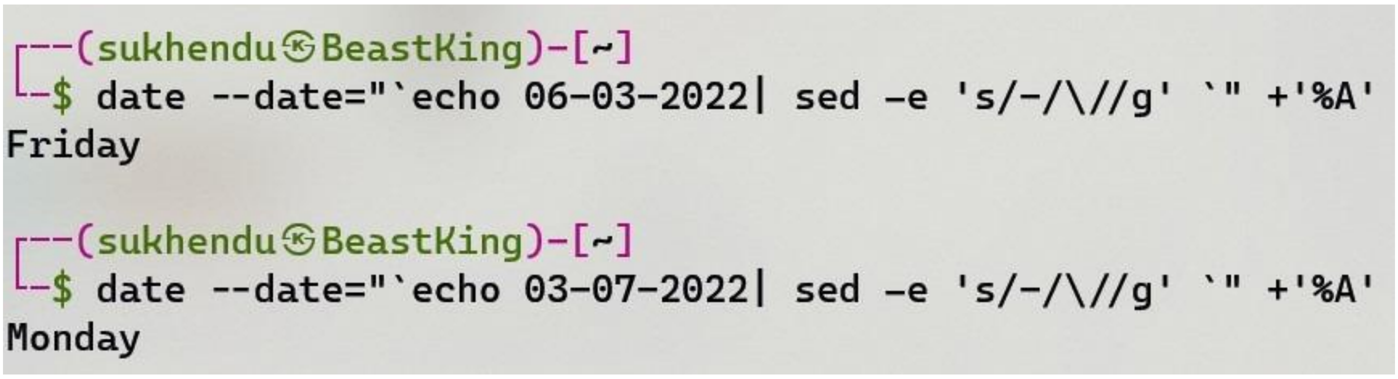
b. With a user-specified date, display only the day of the week (e.g., Tuesday).

a.



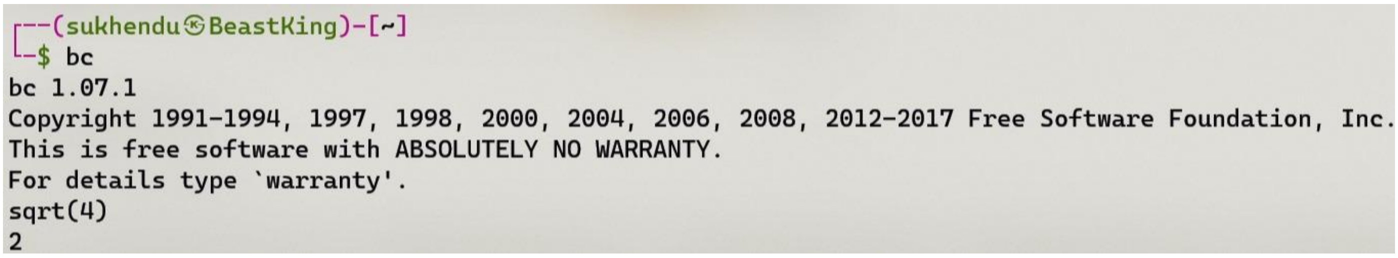
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ date
Sun Mar  6 04:40:58 PM IST 2022

┌──(sukhendu㉿BeastKing)-[~]
└─$ █
```

b.



```
┌──(sukhendu㉿BeastKing)-[~]
└─$ date --date="`echo 06-03-2022| sed -e 's/-/\//g' `" +'%A'
Friday

┌──(sukhendu㉿BeastKing)-[~]
└─$ date --date="`echo 03-07-2022| sed -e 's/-/\//g' `" +'%A'
Monday
```

2. Write the command to find the square root of 4.



```
┌──(sukhendu㉿BeastKing)-[~]
└─$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
sqrt(4)
2
```

Name: Roll No:

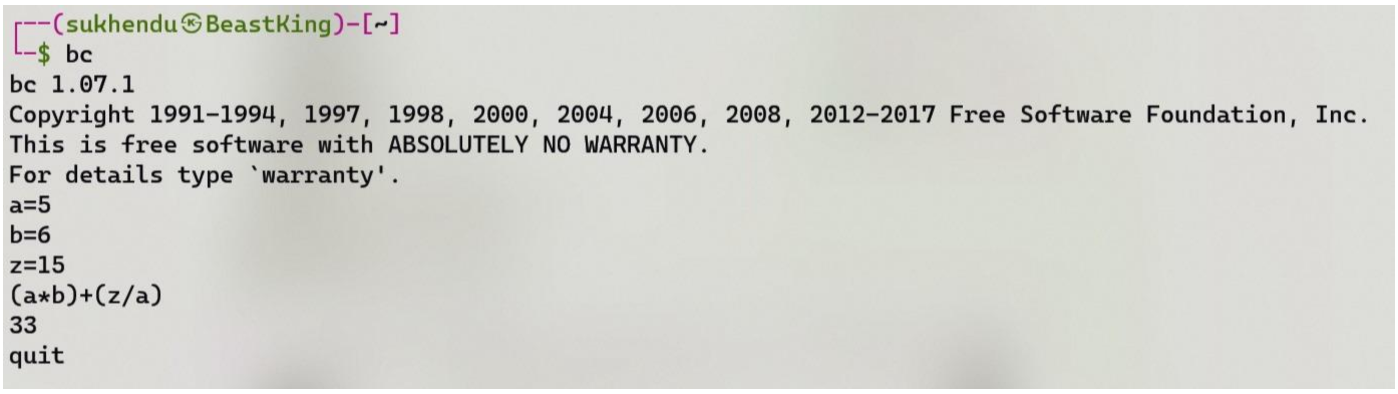
Section: Year:

3. Show how we can calculate the following expression in the terminal of UNIX

A=5, b=6, z=15

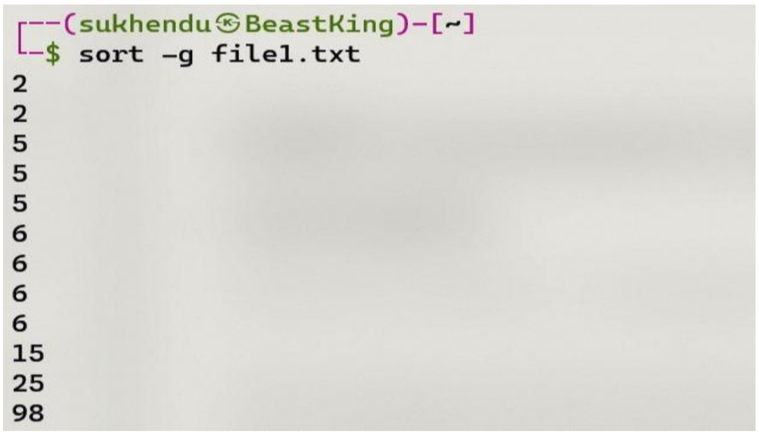
Total = (A*b) + (z/A)

Display the Total.

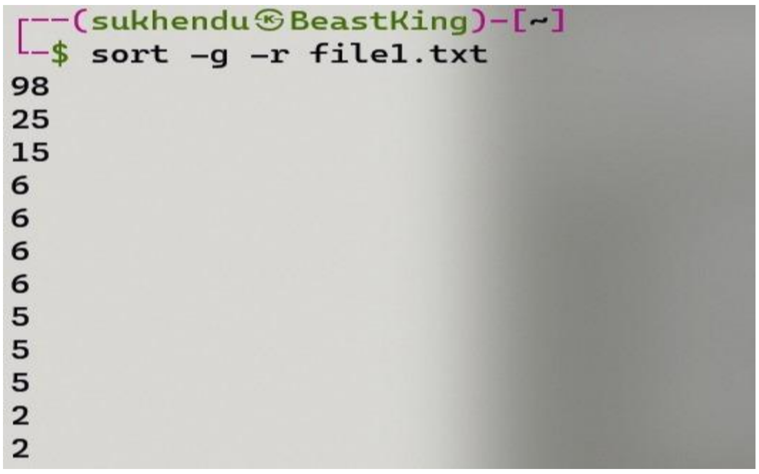


```
┌──(sukhendu㉿BeastKing)-[~]
└─$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
a=5
b=6
z=15
(a*b)+(z/a)
33
quit
```

4. How can we sort a list of numbers in a file (both ascending and descending order)?



```
┌──(sukhendu㉿BeastKing)-[~]
└─$ sort -g file1.txt
2
2
5
5
5
6
6
6
6
15
25
98
```



```
┌──(sukhendu㉿BeastKing)-[~]
└─$ sort -g -r file1.txt
98
25
15
6
6
6
6
5
5
5
2
2
```

5. Create the file student.dat as follows:

Roll | Name | Dept | Year

105 | Anik | CSE | 1st

101 | Debesh | CSE | 2nd

108 | Aniket | IT | 1st

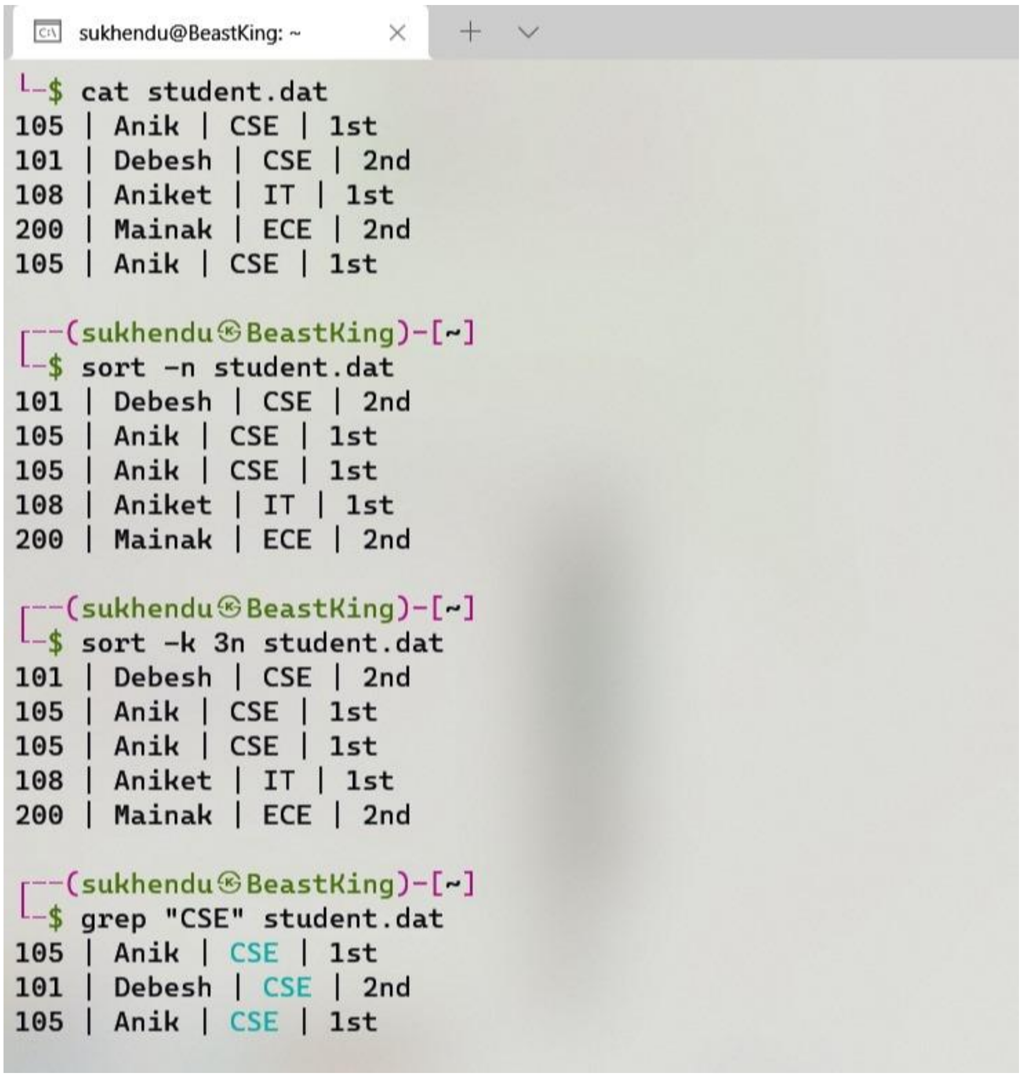
200 | Mainak | ECE | 2nd

105 | Anik | CSE | 1st

a. Sort the data according to Roll.

b. Sort the data according to Dept.

c. Show only the records of students from the CSE Dept.

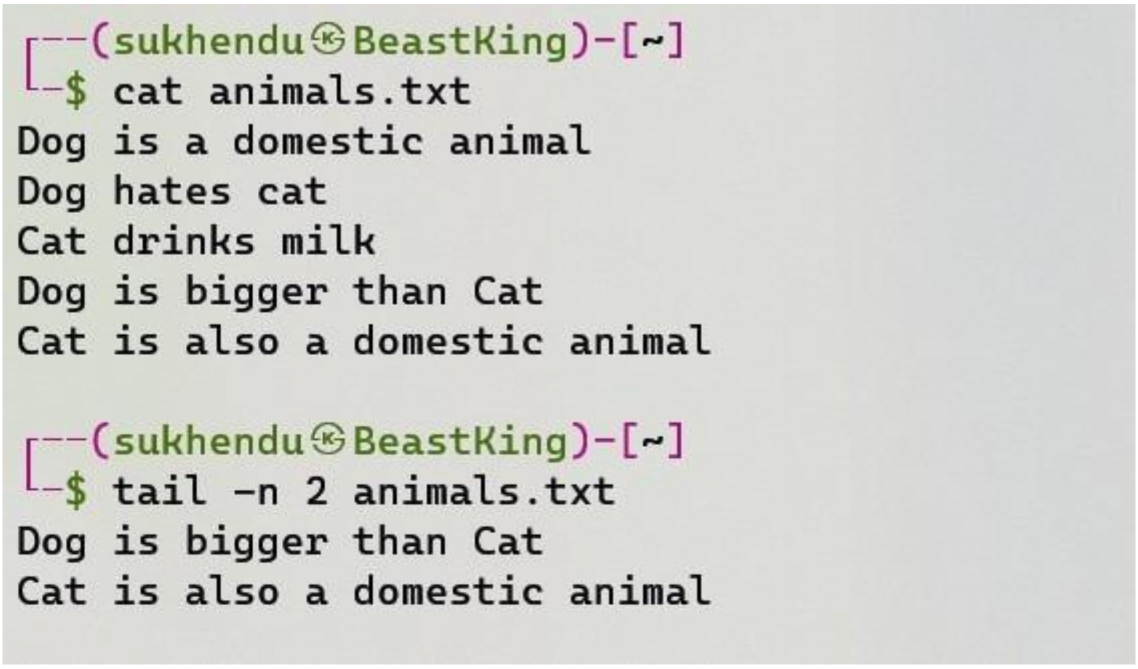


```
└─$ cat student.dat
105 | Anik | CSE | 1st
101 | Debesh | CSE | 2nd
108 | Aniket | IT | 1st
200 | Mainak | ECE | 2nd
105 | Anik | CSE | 1st

┌──(sukhendu㉿BeastKing)-[~]
└─$ sort -n student.dat
101 | Debesh | CSE | 2nd
105 | Anik | CSE | 1st
105 | Anik | CSE | 1st
108 | Aniket | IT | 1st
200 | Mainak | ECE | 2nd

┌──(sukhendu㉿BeastKing)-[~]
└─$ sort -k 3n student.dat
101 | Debesh | CSE | 2nd
105 | Anik | CSE | 1st
105 | Anik | CSE | 1st
108 | Aniket | IT | 1st
200 | Mainak | ECE | 2nd

┌──(sukhendu㉿BeastKing)-[~]
└─$ grep "CSE" student.dat
105 | Anik | CSE | 1st
101 | Debesh | CSE | 2nd
105 | Anik | CSE | 1st
```

Name: Roll No:

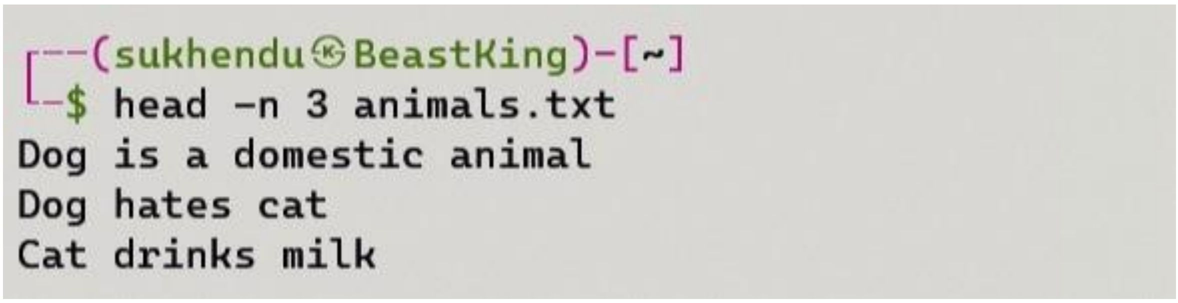
Section: Year:

6. Show the last 2 lines of the file animals.txt.



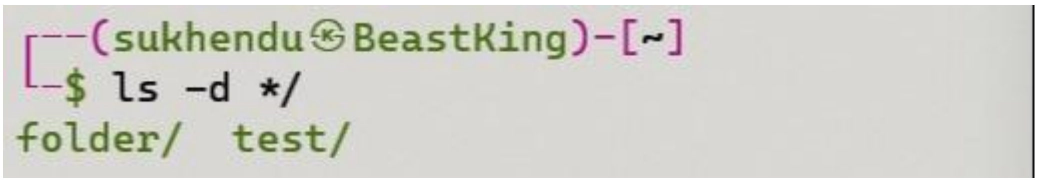
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ cat animals.txt
Dog is a domestic animal
Dog hates cat
Cat drinks milk
Dog is bigger than Cat
Cat is also a domestic animal

┌──(sukhendu㉿BeastKing)-[~]
└─$ tail -n 2 animals.txt
Dog is bigger than Cat
Cat is also a domestic animal
```

7. Show the first 3 lines of the file animals.txt.



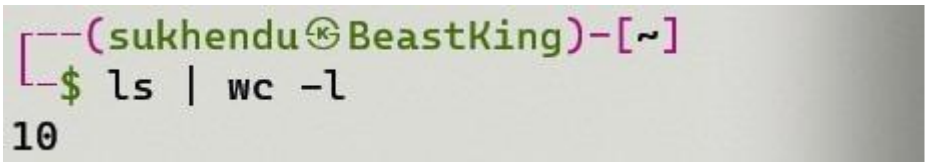
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ head -n 3 animals.txt
Dog is a domestic animal
Dog hates cat
Cat drinks milk
```

8. (Re-Visit) List only the directory files in your current directory.



```
┌──(sukhendu㉿BeastKing)-[~]
└─$ ls -d */
folder/  test/
```

9. Count the number of directories in your current directory.



```
┌──(sukhendu㉿BeastKing)-[~]
└─$ ls | wc -l
10
```

Name: Roll No:

Section: Year:

10.

Dog is a domestic animal

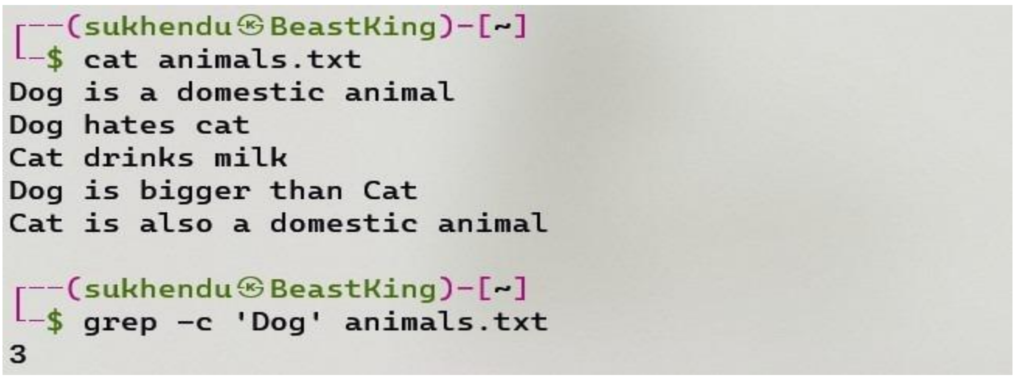
Dog hates cat

Cat drinks milk

Dog is bigger than Cat

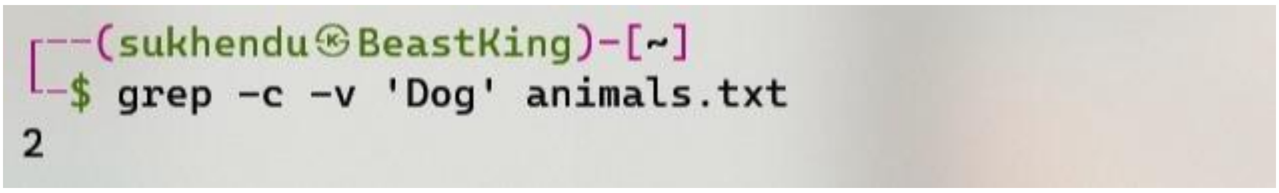
Cat is also a domestic animal

a. Find the total number of lines contains the word ‘Dog’ in animals.txt.



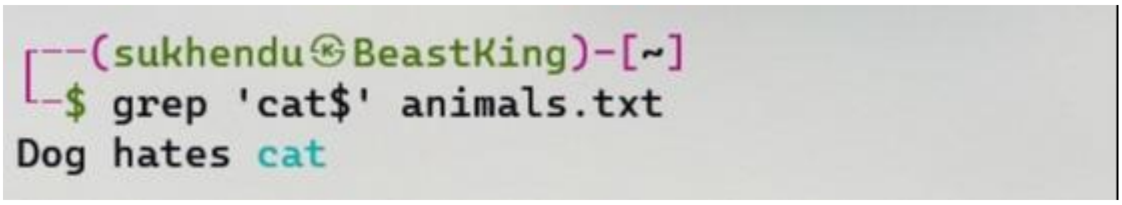
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ cat animals.txt
Dog is a domestic animal
Dog hates cat
Cat drinks milk
Dog is bigger than Cat
Cat is also a domestic animal

┌──(sukhendu㉿BeastKing)-[~]
└─$ grep -c 'Dog' animals.txt
3
```

b. Also find the total number of lines does not contain the word ‘Dog’ in animals.txt.



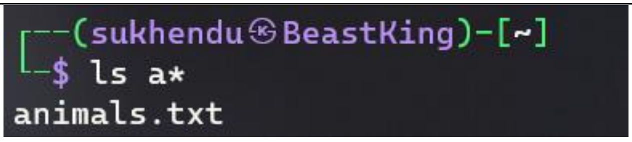
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ grep -c -v 'Dog' animals.txt
2
```

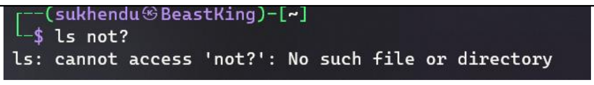
c. Display the lines in animals.txt that end with the word 'cat'.

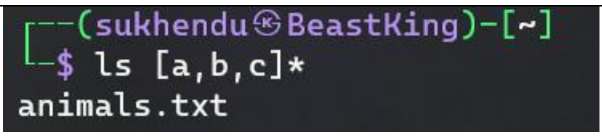


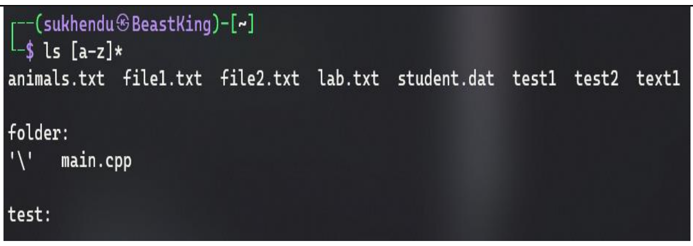
```
┌──(sukhendu㉿BeastKing)-[~]
└─$ grep 'cat$' animals.txt
Dog hates cat
```

Command	**Description**	**Syntax**	**Sample Output**

* wildcard	The * (asterisk) metacharacter is used to match any and all characters.	$ ls [Any character]*	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ ls a*`
			`animals.txt`

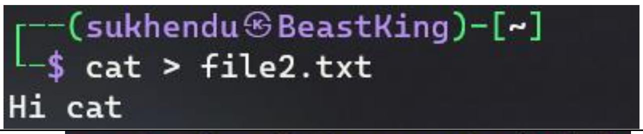
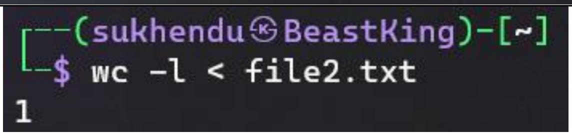
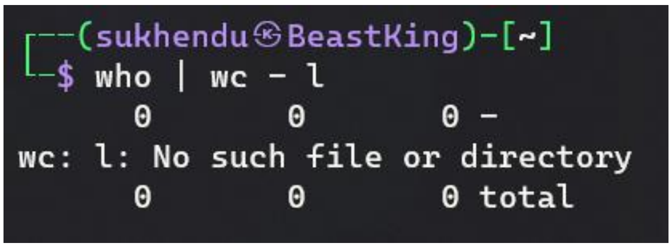
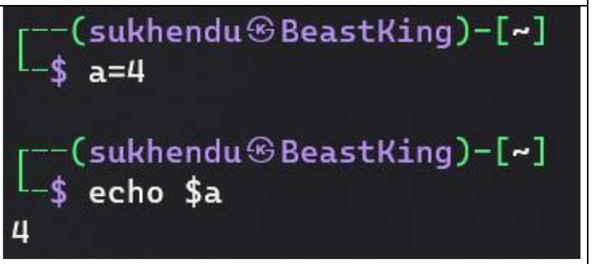
? question mark	The ? (question mark) metacharacter is used to match a single character in a filename.	$ ls [Beginning characters] ?	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ ls not?`
			`ls: cannot access 'not?': No such file or directory`

[ ] brackets	Brackets ([...]) are used to match a set of specified characters. A comma separates each character within the set.	$ ls [Character s to be matched]*	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ ls [a,b,c]*`
			`animals.txt`

- hyphen	Using the - (hyphen) metacharacter within [] (brackets) is used to match a specified range of characters.	$ ls [Character s range]*	
			`┌──(sukhendu㉿BeastKing)-[~]`
			`└─$ ls [a-z]*`
			`animals.txt  file1.txt  file2.txt  lab.txt  student.dat  test1  test2  text1`
			
			`folder:`
			`'\'   main.cpp`
			
			`test:`

Name: Roll No:

Section: Year:

> redirection	Redirect the standard output to replace the current content.	$[comman d whose output is to be copied] > file where the output is to be kept	 `┌──(sukhendu㉿BeastKing)-[~]` `└─$ cat > file2.txt` `Hi cat`
< redirection	Redirect the standard input to a particular command.	$[comman d] < file from which content is to be redirected	 `┌──(sukhendu㉿BeastKing)-[~]` `└─$ wc -l < file2.txt` `1`
| pipe	Pipe | separates commands to form a pipe.	command_1 | command_2 | command_3 | …. | command_N	 `┌──(sukhendu㉿BeastKing)-[~]` `└─$ who | wc - l` `0 0 0 -` `wc: l: No such file or directory` `0 0 0 total`
$ (system) variable	Indicates that the following text is the name of a shell (environment) variable whose value is to be used.	$NAME	 `┌──(sukhendu㉿BeastKing)-[~]` `└─$ a=4` `┌──(sukhendu㉿BeastKing)-[~]` `└─$ echo $a` `4`