Assignment2

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

Command	Description	Syntax	Sample Output
* wildcard	The * (asterisk) metacharacter is used to match any and all characters.	$ ls [Any character]*	ls N* NTUSER.DAT NTUSER.DAT{9d285f09-8fba-11ec-b1c6-91c5b9360fe9}.TMContainer00000000000000000001.regtrans-ms
? question mark	The ? (question mark) metacharacter is used to match a single character in a filename.	$ ls [Beginning characters]?	ls doc?.txt doc1.txt doc2.txt
[ ] brackets	Brackets ([...]) are used to match a set of specified characters. A comma separates each character within the set.	$ ls [Characters to be matched]*	ls [d,o,c]* desktop.ini doc1.txt doc2.txt
- hyphen	Using the - (hyphen) metacharacter within [] (brackets) is used to match a specified range of characters.	$ ls [Characters range]*	ls [c-d]* desktop.ini doc1.txt doc2.txt
> redirection	Redirect the standard output to replace the current content.	$[command whose output is to be copied] > file where the output is to be kept	who > doc1.txt cat doc1.txt shreyoshi16 pts/0 2022-02-26 21:22
< redirection	Redirect the standard input to a particular command.	$[command] < file from which content is to be redirected	wc -l < doc1.txt 1

| pipe	Pipe | seperates commands to form a pipe.	`command_1 | command_2 | command_3 | .... | command_N`	who | wc -l 1
$ (system) variable	Indicates that the following text is the name of a shell (environment) variable whose value is to be used.	`$NAME`	a=5 echo $a 5
cal	Shows the current calender in the terminal. It can be used with several options.	`cal [ [ month ] year]`	Cal February 2022 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
date	Shows the date and time to the nearest second.	`date [OPTION]... [+FORMAT]`	Date Sat Feb 26 22:34:07 IST 2022
cmp	It compares two files of any type and writes the results to the standard output. By default, cmp is silent if the files are the same; if they differ, the byte and line number at which the first difference occurred is reported.	`cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]`	cmp doc1.txt doc2.txt doc1.txt doc2.txt differ: byte 1, line 1

comm	The comm command compares two sorted files line by line. With no options, comm produces three-column output. Column one contains lines unique to FILE1, column two contains lines unique to FILE2, and column three contains lines common to both files.	**`$comm [OPTION]... FILE1 FILE2`**	comm doc1.txt doc2.txt 1 2
diff	It tells which lines of one file have to be changed to make two files identical.	**`diff [options] File1 File2`**	diff doc1.txt doc2.txt 1c1 < 1 --- > 2
head	head by default, prints the first 10 lines of each FILE to standard output.	`head [OPTION]... [FILE]...`	head test.txt 1 2 3 4 5 6 7 8 9 10
Tail	Print the last 10 lines of each FILE to standard output.	`tail [OPTION]... [FILE]...`	xxtail test.txt 3 4 5 6 7 8 9 10 11 12

sort	sort is a simple and very useful command which will rearrange the lines in a text file so that they are sorted, numerically and alphabetically.	**$ sort filename.txt**	sort names.txt anushka batman moumita sayantan shreyoshi srinjoy sruti tania vedant
bc	**bc** command is used for command line calculator. It is similar to basic calculator by using which we can do basic mathematical calculations.	**bc [-hlwsqv] [long-options] [file ...]**	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'. 2+2;4+4 4 8
expr	The **expr** command in Unix evaluates a given expression and displays its corresponding output.	$expr expression	expr 12 + 8 20
grep	The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.	**grep [options] pattern [files]**	grep -i "shreyoshi" names.txt shreyoshi

1.

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

```
date +%r
01:23:35 AM
```

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

```
date --date="1 year"
```

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

```
echo "scale=2; sqrt(4)" | bc -l
2.00
```

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.

```
A=5
b=6
z=15
MUL=$(( A * b ))
DIV=$(( z / A ))
RES=$((MUL + DIV))
echo $RES
33
```

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

```
sort -n filename.txt
```

sorts the numbers in ascending order

```
sort -nr filename.txt
```

sorts the number in descending order

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
```

```
cat > student.dat
```

```
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.

```
sort -n -k1 student.dat
```

```
Roll | Name | Dept |Year
101 | Debesh | CSE | 2nd
105 | Anik | CSE | 1st
105 | Anik | CSE | 1st
108 | Aniket | IT | 1st
200 | Mainak | ECE | 2nd
```

b. Sort the data according to Dept.

```
echo "$(tail -n +2 student.dat)" > student.dat
sort -k5 student.dat
```

```
105 | Anik | CSE | 1st
105 | Anik | CSE | 1st
101 | Debesh | CSE | 2nd
200 | Mainak | ECE | 2nd
108 | Aniket | IT | 1st
```

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

```
grep -i "CSE" student.dat
```

```
105 | Anik | CSE | 1st
101 | Debesh | CSE | 2nd
105 | Anik | CSE | 1st
```

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

```
tail -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.

```
head -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.

```
find . -type d
```

```
.
./Testing1
./Testing1/Testing3
```

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

```
find . -type d | wc -l
```

```
3
```

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.

```
grep -i "Dog" animals.txt | wc -l
```

```
3
```

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

```
grep -v "Dog" animals.txt
```

```
Cat drinks milk
Cat is also a domestic animal
```

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

```
grep 'cat$' animals.txt
```

```
Dog hates cat
```