OS_Lab7

K. Ram Mohan

COE19B055

Q1)

AWK command- Explanation:

COE19B055

1. awk : It is a scripting language used for manipulating data & generate reports.

Syntax:

awk options 'criteria {action}' input file > output file

Options:-

-f program-file : Reads the AWK program source from file program-file, instead from command

-F fs : use fs for input field separator

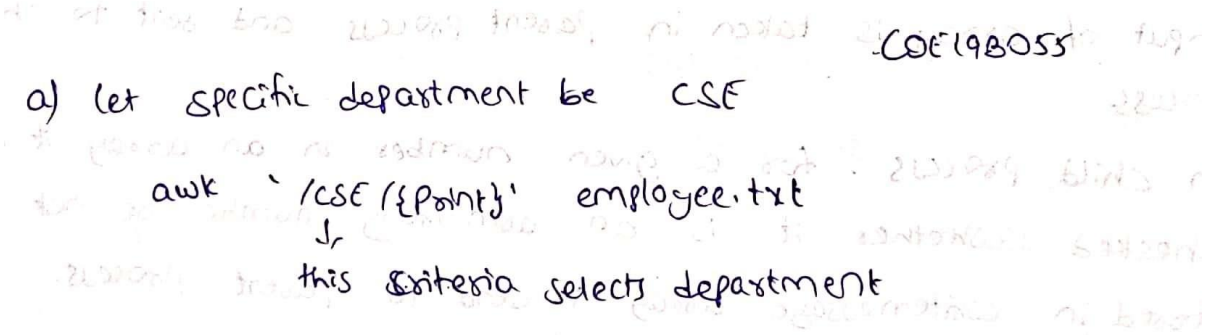
There are also some variable in awk.

- $1, $2, -- represents field1, field2, -- respectively. $0 represents complete line.
- NR - no:of rows
- NF - no:of fields
- FS - field separator
- RS - stores current record separator
- OFS - stores output field separator. Default blank space
- ORS - output record separator. Default new line.

Data in sample text file:

id	Name	Department	Salary	Experience
11	Ram	CSE	35000	2
22	Gopi	ECE	30000	3
33	Anand	ECE	31000	1
44	Gopal	MEC	40000	5
55	Shyam	CSE	20000	0
66	Sam	ECE	100000	2
77	Jathin	MEC	45000	1
88	Naresh	ECE	10000	0
99	Ramesh	CIVIL	36000	2
100	Suresh	CSE	15000	1

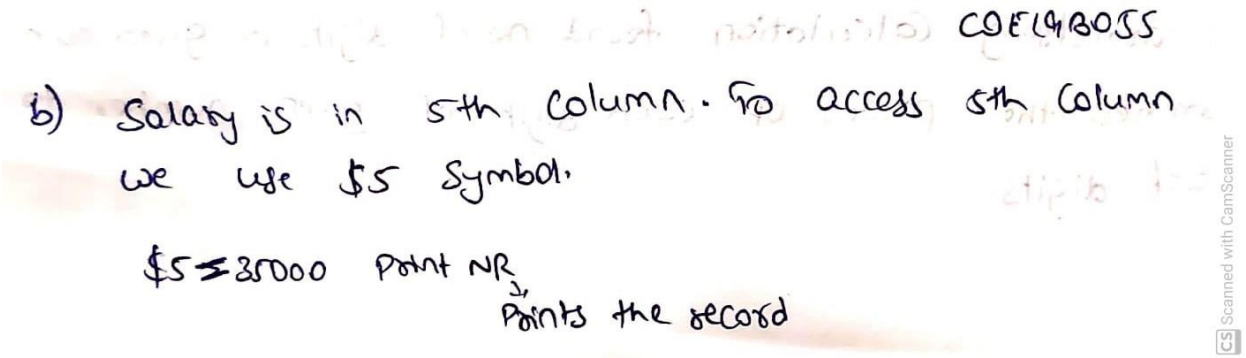
a) Printing details of specific department. Here department taken as CSE



Output:

```
ram@ram:~/Documents/OS$ awk '{if($3=="CSE") print $0}' employee.txt
11      Ram     CSE     35000   2
55      Shyam   CSE     20000   0
100     Suresh  CSE     15000   1
```

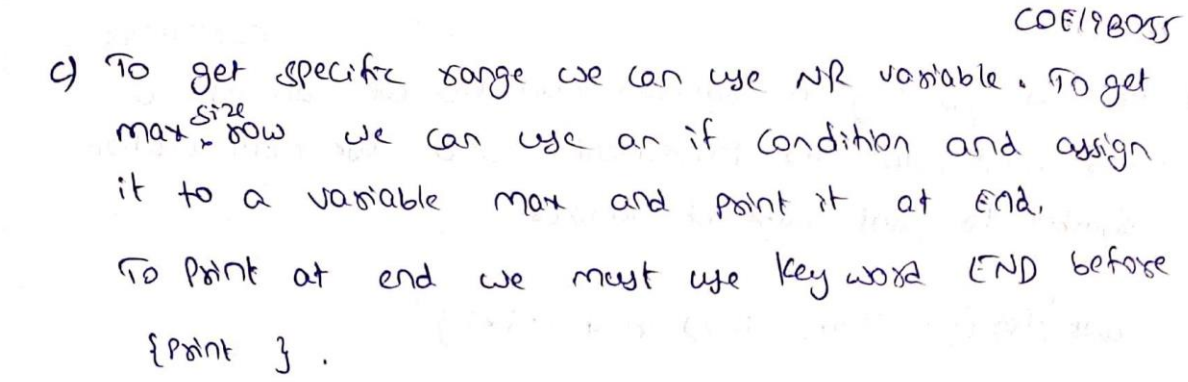
b) Printing records with salary equal to 35000



Output:

```
ram@ram:~/Documents/OS$ awk '{if($4==35000) print NR"-"$0}' employee.txt
1-11    Ram     CSE     35000   2
```

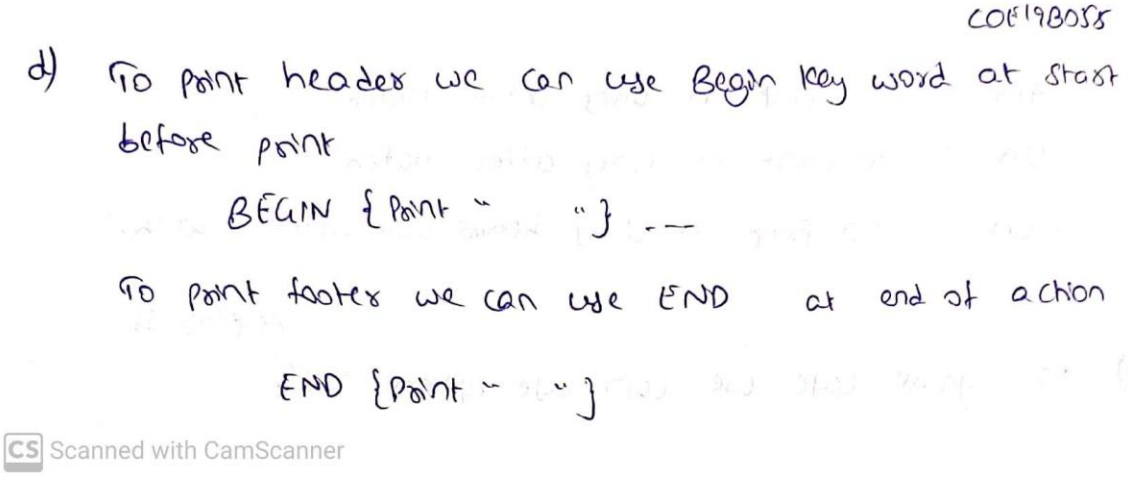
c)Printing max length of record with in a range



Output:

```
ram@ram:~/Documents/OS$ awk 'NR==2, NR==6 {if(length($0)>max) max=length($0)} END {print max}' employee.txt
20
ram@ram:~/Documents/OS$
```

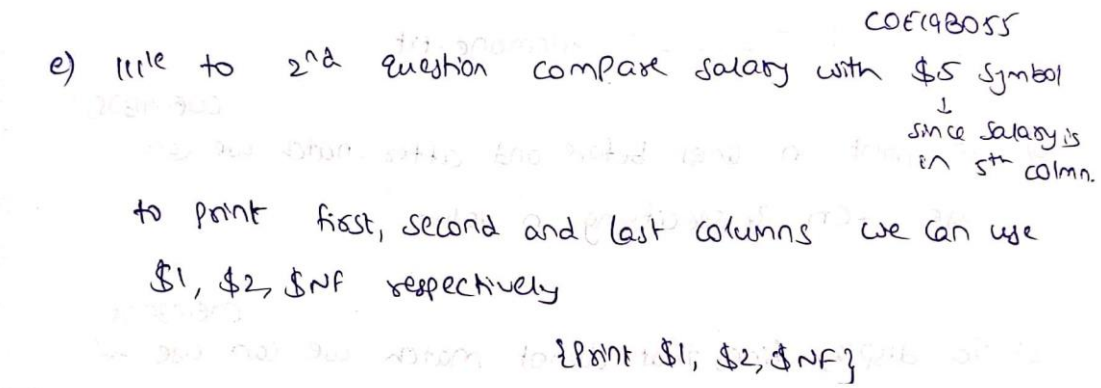
d)Printing data with a header and footer



Output:

```
ram@ram:~/Documents/OS$ awk 'BEGIN {print "Printing employee details"} {print $0} END {print "Ended"}' employ
ee.txt
Printing employee details
11      Ram     CSE     35000   2
22      Gopi    ECE     30000   3
33      Anand   ECE     31000   1
44      Gopal   MEC     40000   5
55      Shyam   CSE     20000   0
66      Sam     ECE     100000  2
77      Jathin  MEC     45000   1
88      Naresh  ECE     10000   0
99      Ramesh  CIVIL   36000   2
100     Suresh  CSE     15000   1
Ended
ram@ram:~/Documents/OS$
```

e) Printing specified columns of records with salary greater than 40000



Output:

```
ram@ram:~/Documents/OS$ awk '{if($4>40000) print $1, $2, $NF}' employee.txt
66 Sam 2
77 Jathin 1
ram@ram:~/Documents/OS$
```

f) Printing average of salary

COE19B055

f) To find Sum we can use statement sum+=$5.
To get average we can divide sum with NR since it has count of records.

Output:

```
ram@ram:~/Documents/OS$ awk '{sum+=$4} END {print sum/NR}' employee.txt
36200
ram@ram:~/Documents/OS$
```

Q2)

a) Printing n random numbers

2) COEIGBOSS

a) To print n random numbers (let n be 10). We can use a for loop in awk, To get random number we can use rand() function.

awk ' for(i=0; i<10; i++) print rand() '

In below picture added some more conditions like Converting rand() given to an integer to make it an integer instead of decimal.

Output:

```
ram@ram:~/Documents/OS$ awk 'BEGIN {n=10; print n," Random numbers:"; for(i=0; i<n; i++) print int(rand()*100)}'
10  Random numbers:
49
23
59
27
84
48
77
17
25
78
ram@ram:~/Documents/OS$
```

b) printing sum of cube of first n even numbers

COEIGBOSS

b) Like to printing n random numbers we can use a for loop with +2 increment and use multiplication symbol to print cube of number

awk {for (i=0; i<10; i=i+2) print i*i*i;} '

Output:

```
ram@ram:~/Documents/OS$ awk 'BEGIN{
n=15; sum=0;
for(i=0; i<n; i=i+2)
{sum=sum+(i*i*i);}
print "Sum of first ", n, " cubes is: ",sum}'
Sum of first  15  cubes is:  6272
ram@ram:~/Documents/OS$
```

c) Printing environmental variables

COE19B055

c) let us Print an environmental variable Path using awk.

We can use option -v to print environmental variable.

We use echo because awk expects an input file to avoid it we use echo.

To print an env variable we must assign it to a local variable in Criteria section of awk and print it.

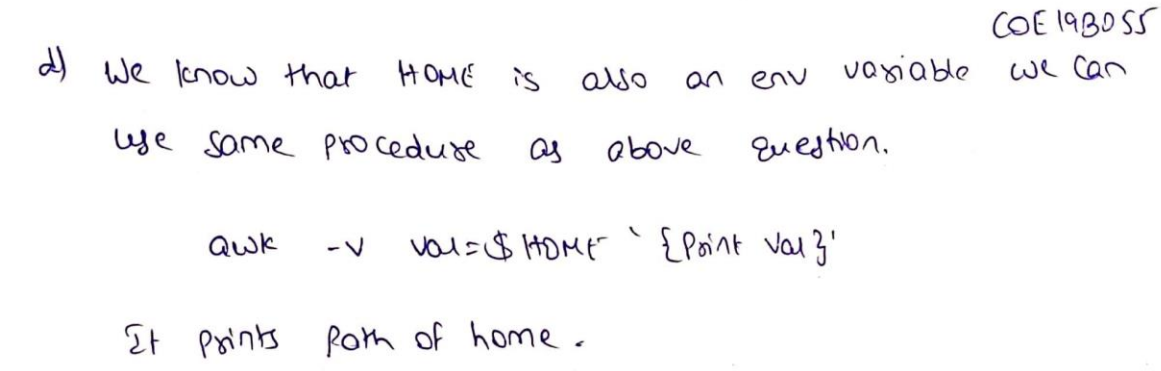
ie, awk -v val=$PATH '{Print val}'

We can also print user defined env variables as shown in below figure.

Output:

```
ram@ram:~/Documents/OS$ value=5
ram@ram:~/Documents/OS$ echo abc |awk -v val=$value '{print val}'
5
ram@ram:~/Documents/OS$ echo abc |awk -v val=$PATH '{print val}'
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc
al/games:/snap/bin
ram@ram:~/Documents/OS$
```

d) Printing home path



Output:

```
ram@ram:~/Documents/OS$ echo abc | awk -v val=$HOME '{print val}'
/home/ram
ram@ram:~/Documents/OS$ awk -v val=$HOME '{print val}'
```

Q3)

file1.txt:

LOREM4 ipsum dolor sit amet, consectetur adipiscing elit.

Aenean luctus tellus consectetur aliquam ultrices.

Duis ultrices quam vitae augue rhoncus tincidunt.

Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.

Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.

Pellentesque lacinia dui eu elit placerat feugiat.

Sed vestibulum orci sed mi viverra porttitor.

Proin semper augue a leo consequat blandit.

Etiam at ex in metus INTERDUM mattis ut vel nisi.

file2.txt:

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Aenean luctus tellus consectetur aliquam ultrices.

Duis ultrices quam vitae augue rhoncus tincidunt.

Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.

Phasellus a sapien dictum, elementum odio vitae, ullamcorper DOLOR.

Pellentesque lacinia dui eu elit placerat feugiat.

Sed vestibulum orci sed mi viverra porttitor.

Proin semper augue a leo consequat blandit.

Etiam at ex in metus interdum mattis ut vel nisi.

Ut LAOREET tellus et mauris iaculis, id tempor leo pretium.

Donec fringilla dui eget elit FEUGIAT consectetur.

Nam at lacus posuere, tristique ante in, pulvinar sapien.

Phasellus rutrum magna id eros porta, sit amet pellentesque mauris scelerisque.

Nunc sed turpis eu nisi mattis laoreet sit amet ac erat.

file3.txt:

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Aenean luctus tellus consectetur aliquam ultrices.

Duis ultrices quam vitae augue rhoncus tincidunt.

Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.

Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.

Pellentesque lacinia dui eu elit placerat feugiat.

Sed vestibulum orci sed mi viverra porttitor.

Proin semper augue a leo consequat blandit.

Etiam at ex in metus interdum mattis ut vel nisi.

Ut laoreet tellus et mauris iaculis, id TEMPOR1 leo pretium.

Donec fringilla dui eget elit feugiat consectetur.

Nam at lacus posuere, tristique ante in, pulvinar sapien.

Phasellus rutrum magna id eros porta, sit amet pellentesque mauris scelerisque.

Nunc sed turpis eu nisi mattis laoreet sit amet ac erat.

In et elit nec tellus pulvinar AUCTOR.

Donec eget velit semper, ornare erat eget, varius dolor.

Vivamus ornare eros ut sollicitudin euismod.

Sed dapibus sem at lorem aliquam, at scelerisque neque tincidunt.

Praesent sit amet arcu vitae purus vestibulum tempus ut at metus.

Morbi auctor turpis quis consequat vulputate.

Grep Explanation:

COELABOSS
KRam Mohan

grep:
helps to filter & search a file

grep [options] "Pattern" filename

-c : displays count of matches (case sensitive)

-i : case insensitive

-l : displays file names that matches patter
(for this file-name is not required - Place *)

-w : match whole word

-o : displays only matched Pattern

-n : Show line number

-v : Print all lines do not match Pattern

^ in pattern : Prints lines that start with pattern

$ in " : Prints lines that end with Pattern

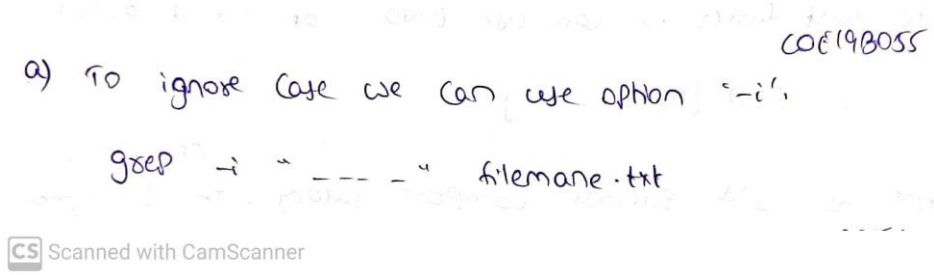
-f : takes patters from file, one per line

-An : To Print n lines before match

-Bn : To Print n lines after match

-Cn : To Print n lines before and after match

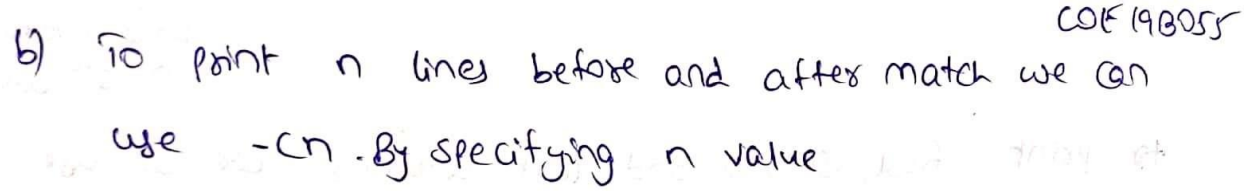
a) Printing matched lines with ignored case



Output:

```
ram@ram:~/Documents/OS$ grep -i "SIT AMET, CONSECTETUR ADIPISCING ELIT" file1.txt file2.txt file3.txt
file1.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file2.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file3.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
ram@ram:~/Documents/OS$
```

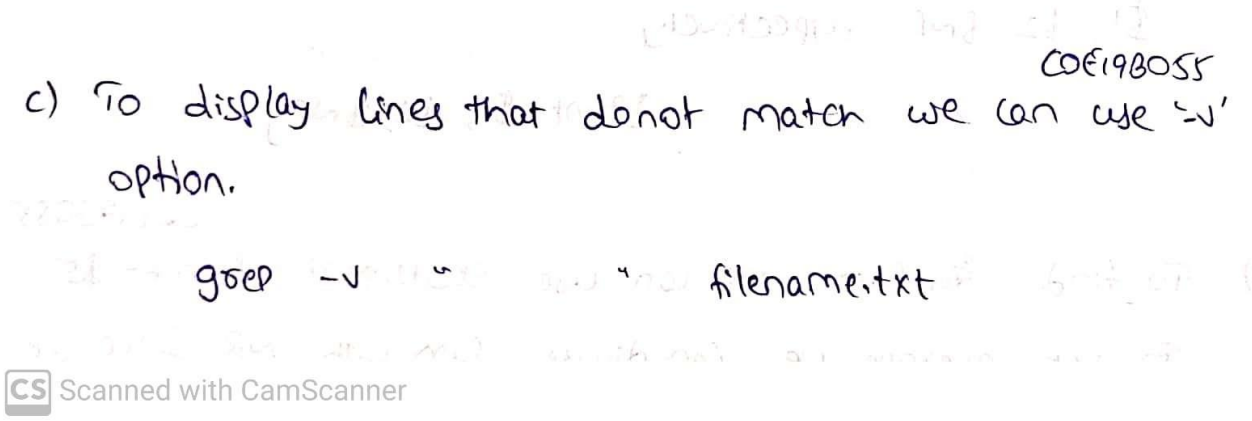
b) Printing n lines before and after match line



Output:

```
ram@ram:~/Documents/OS$ grep -C2 "sed mi viverra porttitor" file1.txt file2.txt file3.txt
file1.txt-Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.
file1.txt-Pellentesque lacinia dui eu elit placerat feugiat.
file1.txt:Sed vestibulum orci sed mi viverra porttitor.
file1.txt-Proin semper augue a leo consequat blandit.
file1.txt-Etiam at ex in metus interdum mattis ut vel nisi.
--
file2.txt-Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.
file2.txt-Pellentesque lacinia dui eu elit placerat feugiat.
file2.txt:Sed vestibulum orci sed mi viverra porttitor.
file2.txt-Proin semper augue a leo consequat blandit.
file2.txt-Etiam at ex in metus interdum mattis ut vel nisi.
--
file3.txt-Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.
file3.txt-Pellentesque lacinia dui eu elit placerat feugiat.
file3.txt:Sed vestibulum orci sed mi viverra porttitor.
file3.txt-Proin semper augue a leo consequat blandit.
file3.txt-Etiam at ex in metus interdum mattis ut vel nisi.
ram@ram:~/Documents/OS$
```

c) Printing lines that do not match with pattern



Output: Complete lines are not shown due to much content

```
ram@ram:~/Documents/OS$ grep -v "sed mi viverra porttitor" file1.txt file2.txt file3.txt
file1.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file1.txt:Aenean luctus tellus consectetur aliquam ultrices.
file1.txt:Duis ultrices quam vitae augue rhoncus tincidunt.
file1.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
file1.txt:Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.
file1.txt:Pellentesque lacinia dui eu elit placerat feugiat.
file1.txt:Proin semper augue a leo consequat blandit.
file1.txt:Etiam at ex in metus interdum mattis ut vel nisi.
file2.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file2.txt:Aenean luctus tellus consectetur aliquam ultrices.
file2.txt:Duis ultrices quam vitae augue rhoncus tincidunt.
file2.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
file2.txt:Phasellus a sapien dictum, elementum odio vitae, ullamcorper dolor.
file2.txt:Pellentesque lacinia dui eu elit placerat feugiat.
file2.txt:Proin semper augue a leo consequat blandit.
file2.txt:Etiam at ex in metus interdum mattis ut vel nisi.
file2.txt:Ut laoreet tellus et mauris iaculis, id tempor leo pretium.
file2.txt:Donec fringilla dui eget elit feugiat consectetur.
file2.txt:Nam at lacus posuere, tristique ante in, pulvinar sapien.
file2.txt:Phasellus rutrum magna id eros porta, sit amet pellentesque mauris scelerisque
file2.txt:Nunc sed turpis eu nisi mattis laoreet sit amet ac erat.
file3.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file3.txt:Aenean luctus tellus consectetur aliquam ultrices.
file3.txt:Duis ultrices quam vitae augue rhoncus tincidunt.
file3.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
```

d) Displaying text in color

d) To display color we can use --color[=when] option

when is 'always', or 'never', or 'auto'

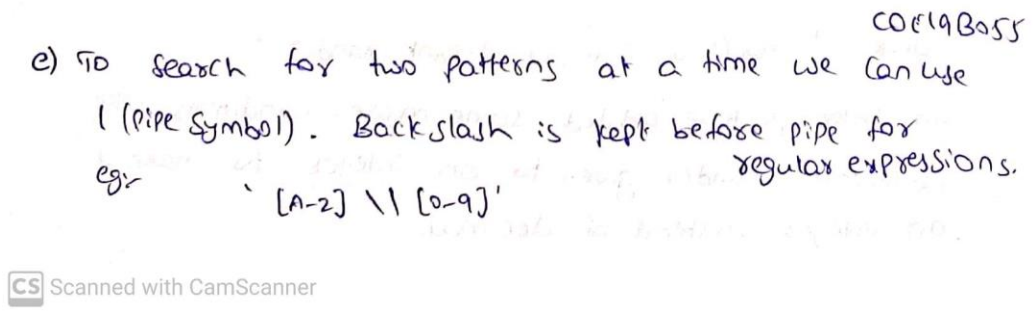
always : } gives a color
auto :

never : doesnot give a color

Output:

```
ram@ram:~/Documents/OS$ grep --color=never -i "SIT AMET, CONSECTETUR ADIPISCING ELIT" file1.txt file2.t
xt file3.txt
file1.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file2.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
file3.txt:Lorem ipsum dolor sit amet, consectetur adipiscing elit.
ram@ram:~/Documents/OS$ grep --color=auto -i "eu consectetur lorem ultrices" file1.txt file2.txt file3.
txt
file1.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
file2.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
file3.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
ram@ram:~/Documents/OS$ grep --color=always -i "eu consectetur lorem ultrices" file1.txt file2.txt file
3.txt
file1.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
file2.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
file3.txt:Aliquam ultrices enim eget sapien tempor, eu consectetur lorem ultrices.
ram@ram:~/Documents/OS$
```

e) Printing patterns match with [A-Z] and [0-9]



Output:

```
ram@ram:~/Documents/OS$ grep -o '[A-Z]*\|[0-9]' file1.txt file2.txt file3.txt
file1.txt:LOREM
file1.txt:4
file1.txt:A
file1.txt:D
file1.txt:A
file1.txt:P
file1.txt:P
file1.txt:S
file1.txt:P
file1.txt:E
file1.txt:INTERDUM
file2.txt:L
file2.txt:A
file2.txt:D
file2.txt:A
file2.txt:P
file2.txt:DOLOR
file2.txt:P
file2.txt:S
file2.txt:P
file2.txt:E
file2.txt:U
file2.txt:LAOREET
file2.txt:D
file2.txt:FEUGIAT
file2.txt:N
file2.txt:P
file2.txt:N
file3.txt:L
```