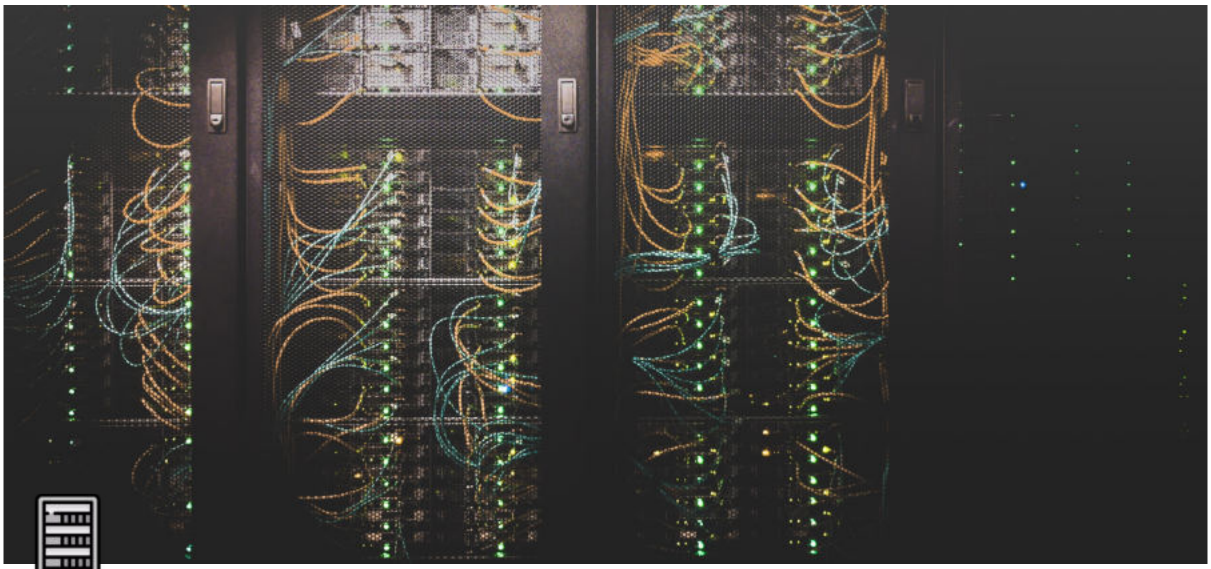


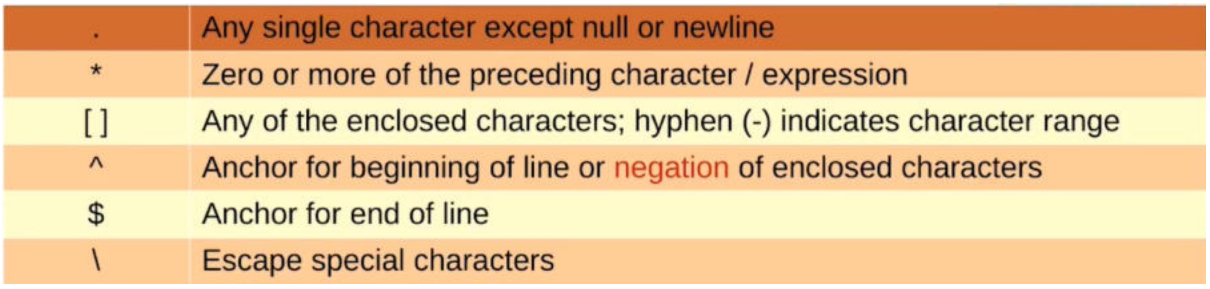
Regex

Type	📒 Lecture
Date	@February 14, 2022
Lecture #	?
Lecture URL	https://youtube.com/playlist?list=PL4cUxeGkcC9g6m_6Sld9Q4jzqdqHd2HiD
Notion URL	https://21f1003586.notion.site/Regex-e0c9ddcaa9364292a0a9bace37ffc948
Week #	4

Usage

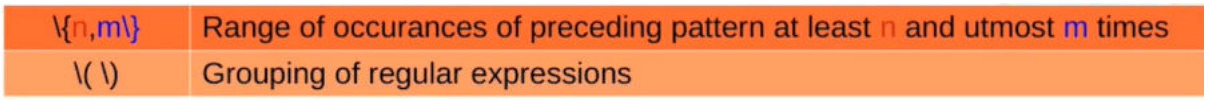
- `grep ‘pattern’ filename`
- `command | grep ‘pattern’`
- Default engine: BRE
- Switch to use ERE:
  - `egrep ‘pattern’ filename`
  - `grep -E ‘pattern’ filename`

Special characters (BRE & ERE)



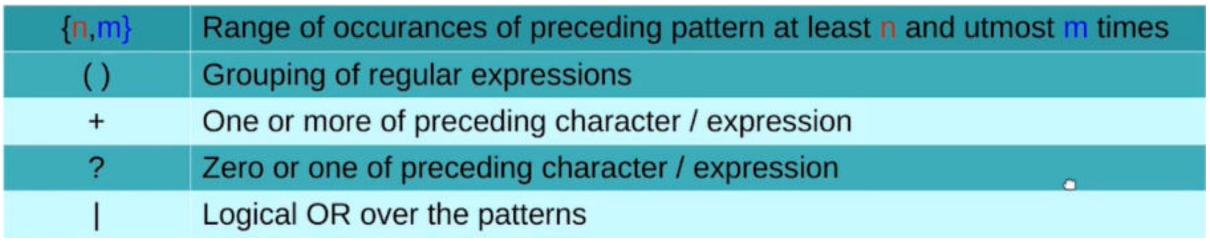
.	Any single character except null or newline
*	Zero or more of the preceding character / expression
[ ]	Any of the enclosed characters; hyphen (-) indicates character range
^	Anchor for beginning of line or negation of enclosed characters
$	Anchor for end of line
\	Escape special characters

Special characters (BRE)



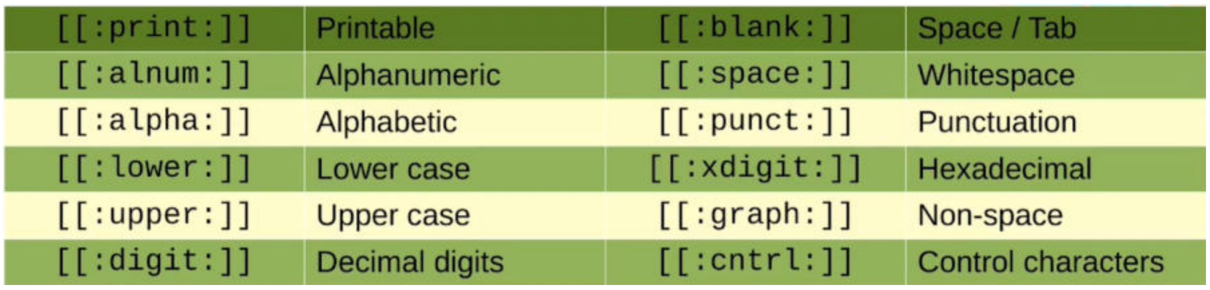
\{n,m\}	Range of occurances of preceding pattern at least n and utmost m times
\(\)	Grouping of regular expressions

Special characters (ERE)



{n,m}	Range of occurances of preceding pattern at least n and utmost m times
( )	Grouping of regular expressions
+	One or more of preceding character / expression
?	Zero or one of preceding character / expression
|	Logical OR over the patterns

Character classes

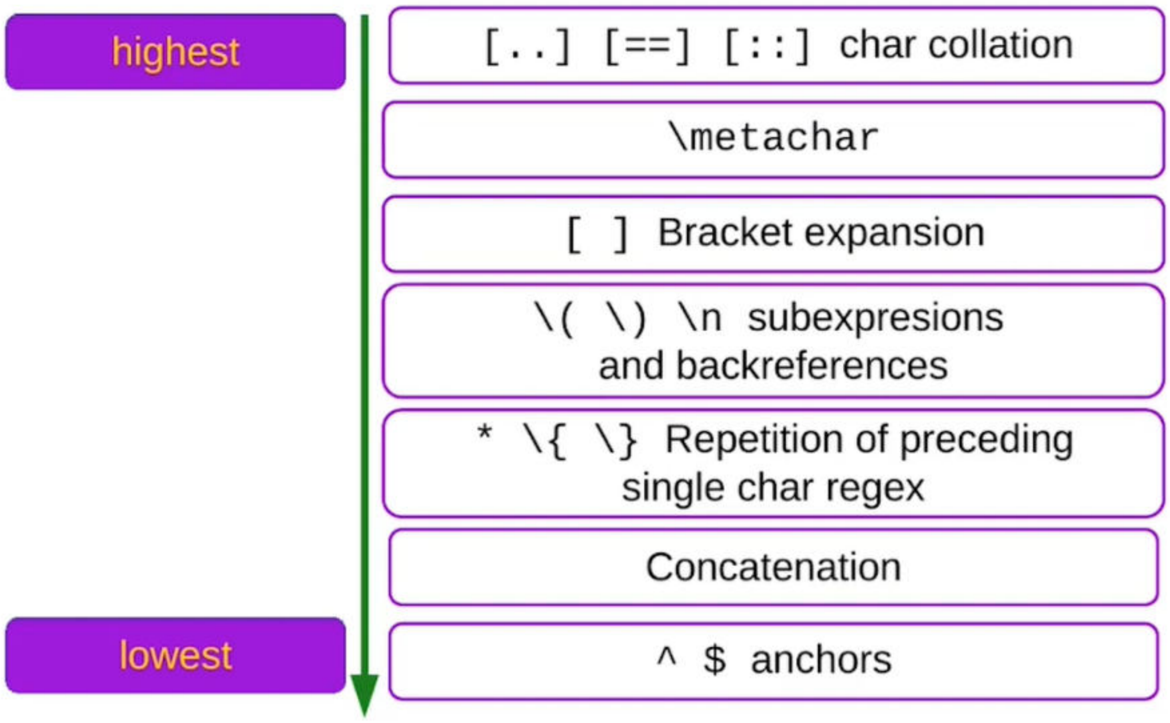


`[[:print:]]`	Printable	`[[:blank:]]`	Space / Tab
`[[:alnum:]]`	Alphanumeric	`[[:space:]]`	Whitespace
`[[:alpha:]]`	Alphabetic	`[[:punct:]]`	Punctuation
`[[:lower:]]`	Lower case	`[[:xdigit:]]`	Hexadecimal
`[[:upper:]]`	Upper case	`[[:graph:]]`	Non-space
`[[:digit:]]`	Decimal digits	`[[:cntrl:]]`	Control characters

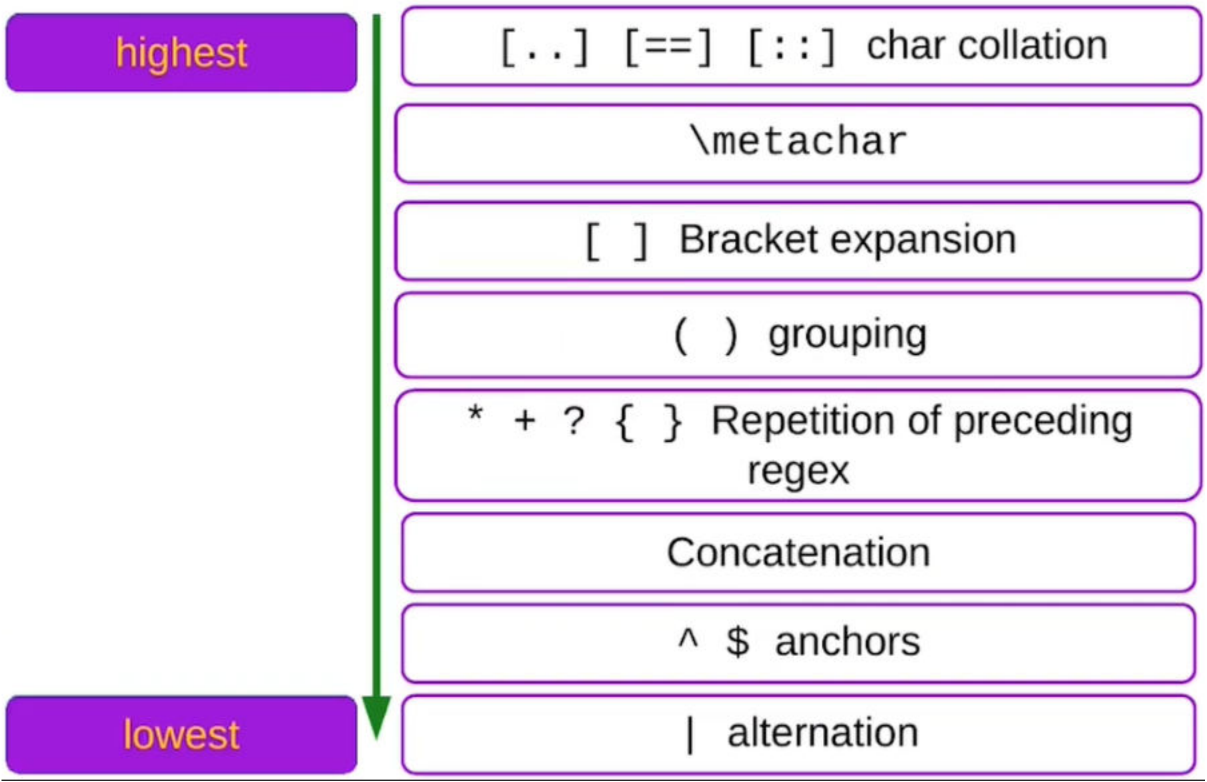
Backreferences

- `\1` through `\9`
- `\n` matches whatever was matched by the `n` th earlier paranthesized sub-expression
- A line with two occurences of *"hello"* will be matched using: `\(hello\).*\1`

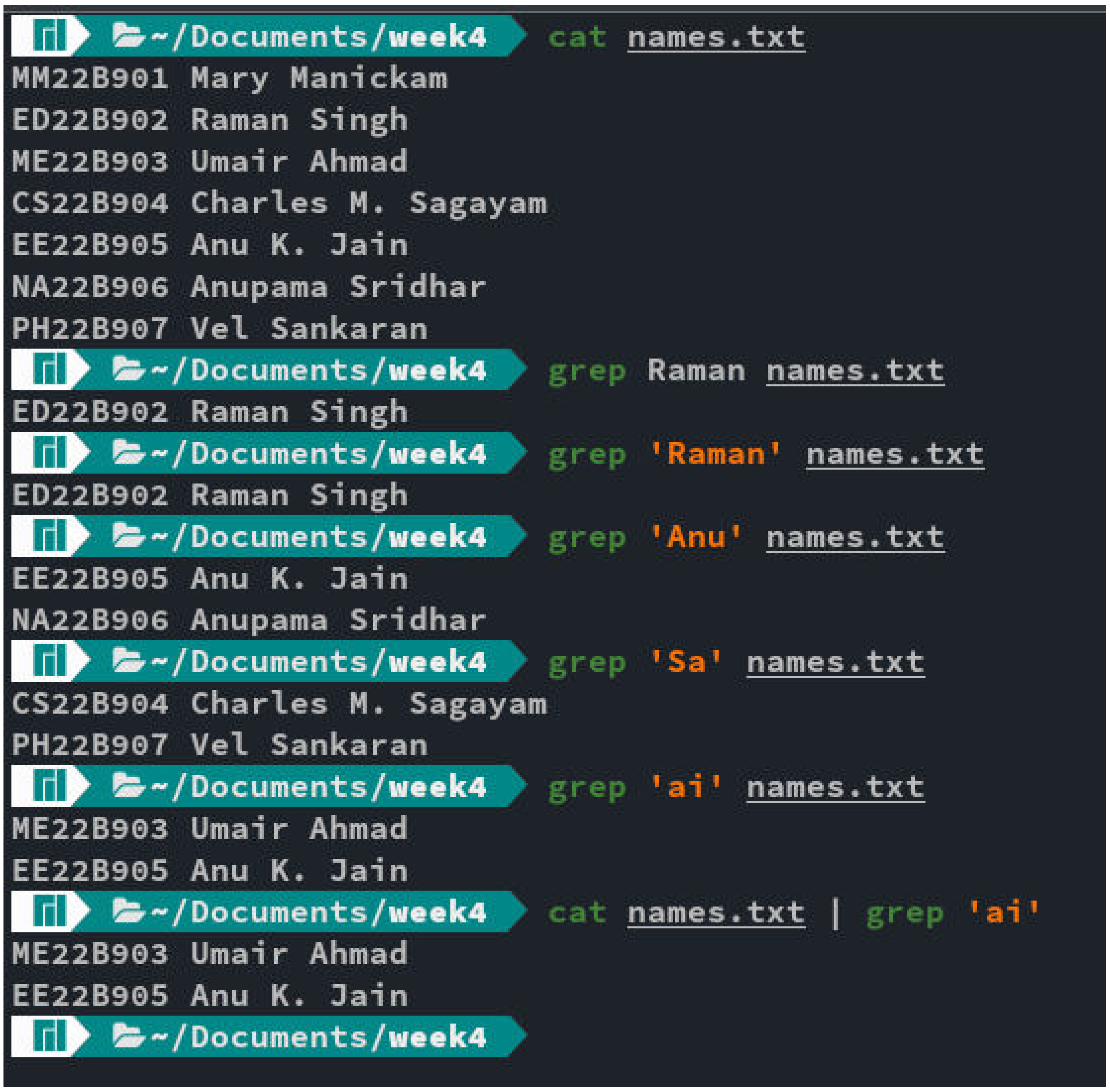
BRE operator precedence


highest
lowest
[..] [==] [::] char collation
\metachar
[ ] Bracket expansion
\(\) \n subexpresions and backreferences
* \{ \} Repetition of preceding single char regex
Concatenation
^ $ anchors

ERE operator precedence



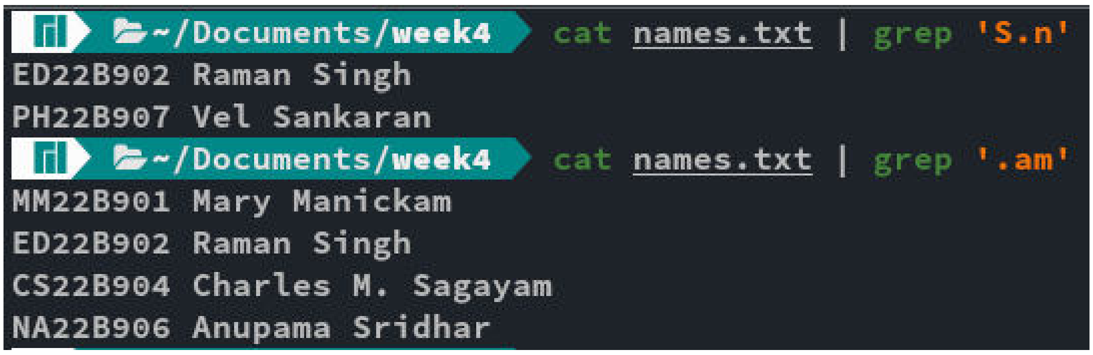
Uses of `grep`



```
~/Documents/week4  cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
~/Documents/week4  grep Raman names.txt
ED22B902 Raman Singh
~/Documents/week4  grep 'Raman' names.txt
ED22B902 Raman Singh
~/Documents/week4  grep 'Anu' names.txt
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
~/Documents/week4  grep 'Sa' names.txt
CS22B904 Charles M. Sagayam
PH22B907 Vel Sankaran
~/Documents/week4  grep 'ai' names.txt
ME22B903 Umair Ahmad
EE22B905 Anu K. Jain
~/Documents/week4  cat names.txt | grep 'ai'
ME22B903 Umair Ahmad
EE22B905 Anu K. Jain
~/Documents/week4
```

Usage of `.` in the `grep` command

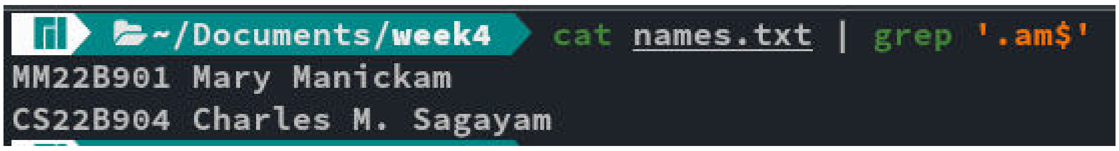
`.` is like a wildcard for a single character



```
~/Documents/week4  cat names.txt | grep 'S.n'
ED22B902 Raman Singh
PH22B907 Vel Sankaran
~/Documents/week4  cat names.txt | grep '.am'
MM22B901 Mary Manickam
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
NA22B906 Anupama Sridhar
```

`$` is used to denote an anchor

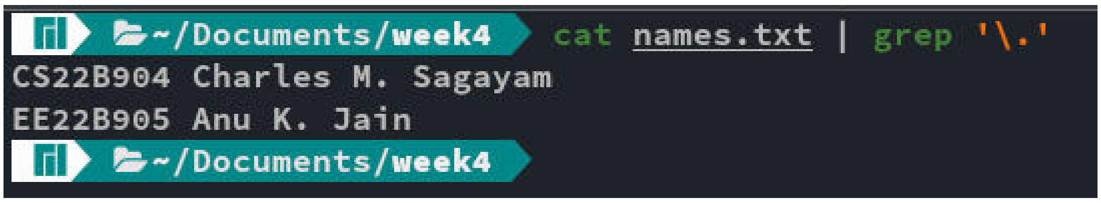
Like a pattern at the end of the line



```
~/Documents/week4  cat names.txt | grep '.am$'
MM22B901 Mary Manickam
CS22B904 Charles M. Sagayam
```

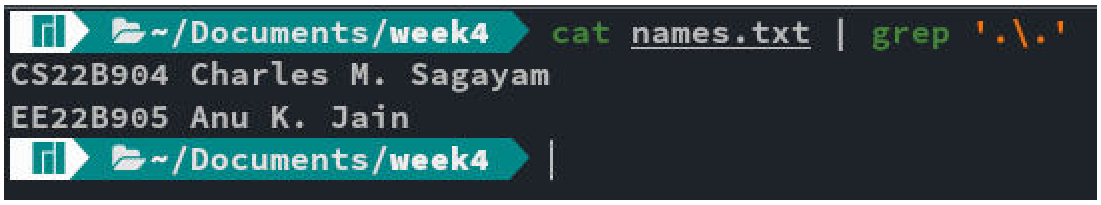
Well what if your name contains a `.`

Then escape it using the `\` character



```
~/Documents/week4  cat names.txt | grep '\.'
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
~/Documents/week4
```

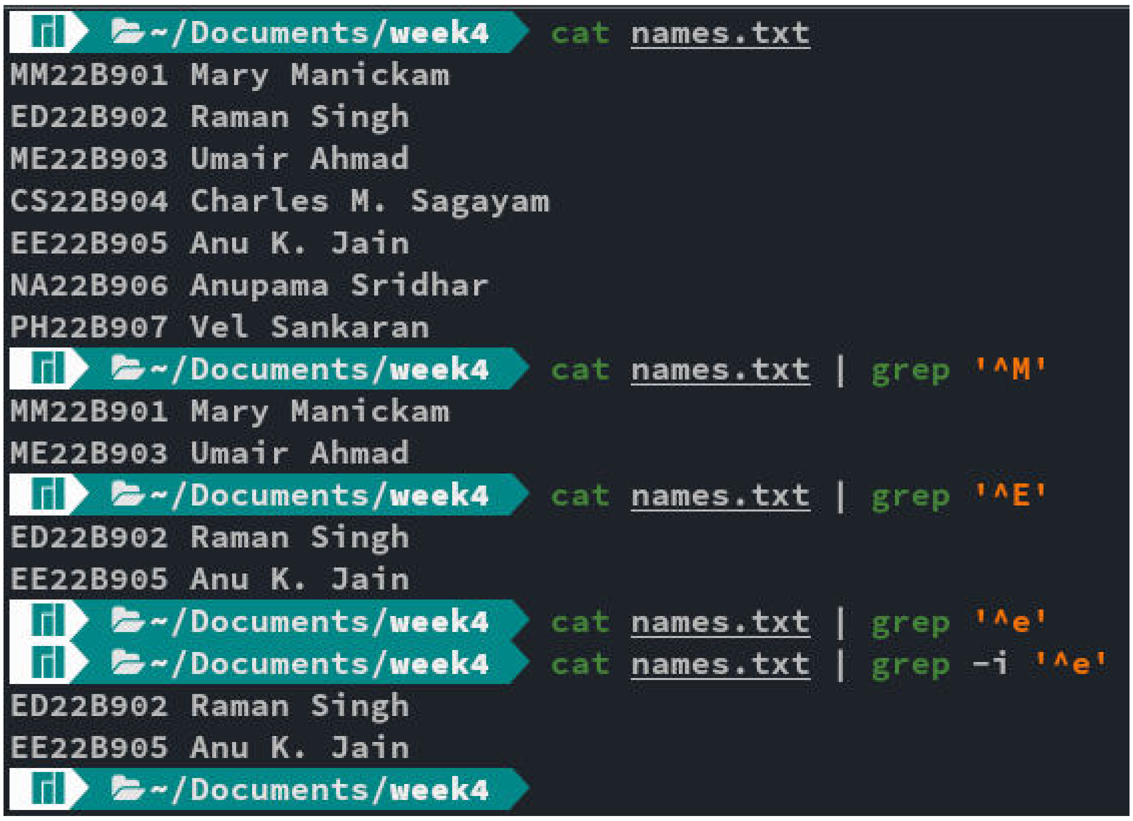
If we want the `.` to be necessarily after a character



```
~/Documents/week4  cat names.txt | grep '.\.'
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
~/Documents/week4  |
```

Match string using anchors at the beginning

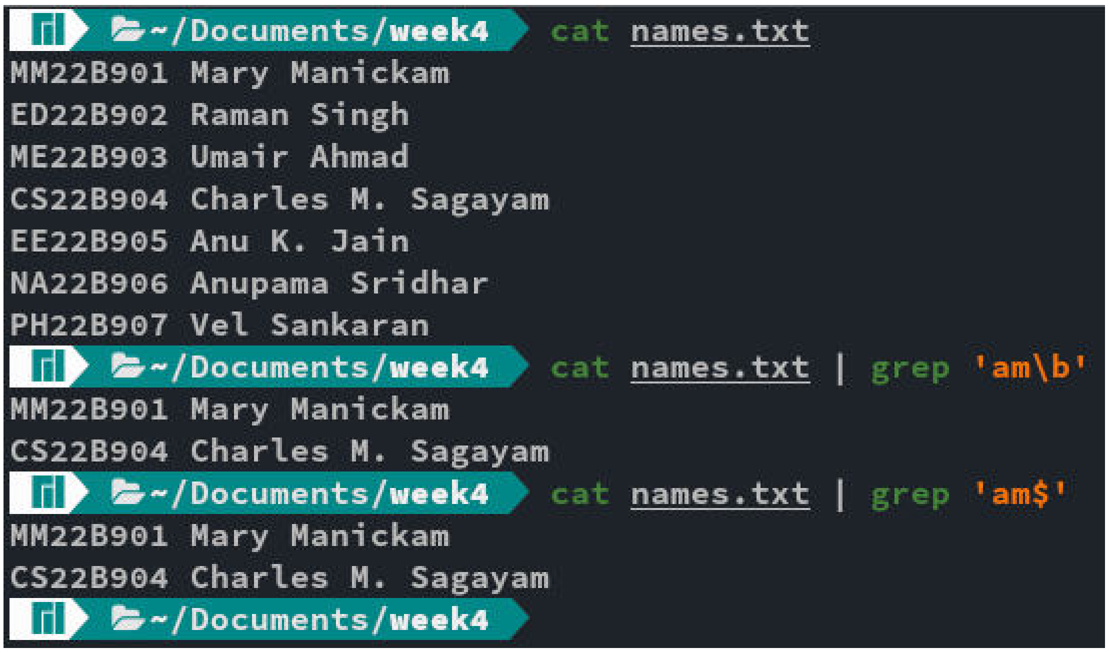
ask `grep` to ignore the case by passing in the `-i` flag


```
 ~/Documents/week4  cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
 ~/Documents/week4  cat names.txt | grep '^M'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
 ~/Documents/week4  cat names.txt | grep '^E'
ED22B902 Raman Singh
EE22B905 Anu K. Jain
 ~/Documents/week4  cat names.txt | grep '^e'
 ~/Documents/week4  cat names.txt | grep -i '^e'
ED22B902 Raman Singh
EE22B905 Anu K. Jain
 ~/Documents/week4
```

Match a pattern at the end of the line, a word boundary one might say

`\b` looks for a word boundary, so that pattern could also occur at the end of a word in the middle of a line

`$` looks for line boundary only, so the pattern occurs at the end of the line


```
 ~/Documents/week4  cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
 ~/Documents/week4  cat names.txt | grep 'am\b'
MM22B901 Mary Manickam
CS22B904 Charles M. Sagayam
 ~/Documents/week4  cat names.txt | grep 'am$'
MM22B901 Mary Manickam
CS22B904 Charles M. Sagayam
 ~/Documents/week4
```

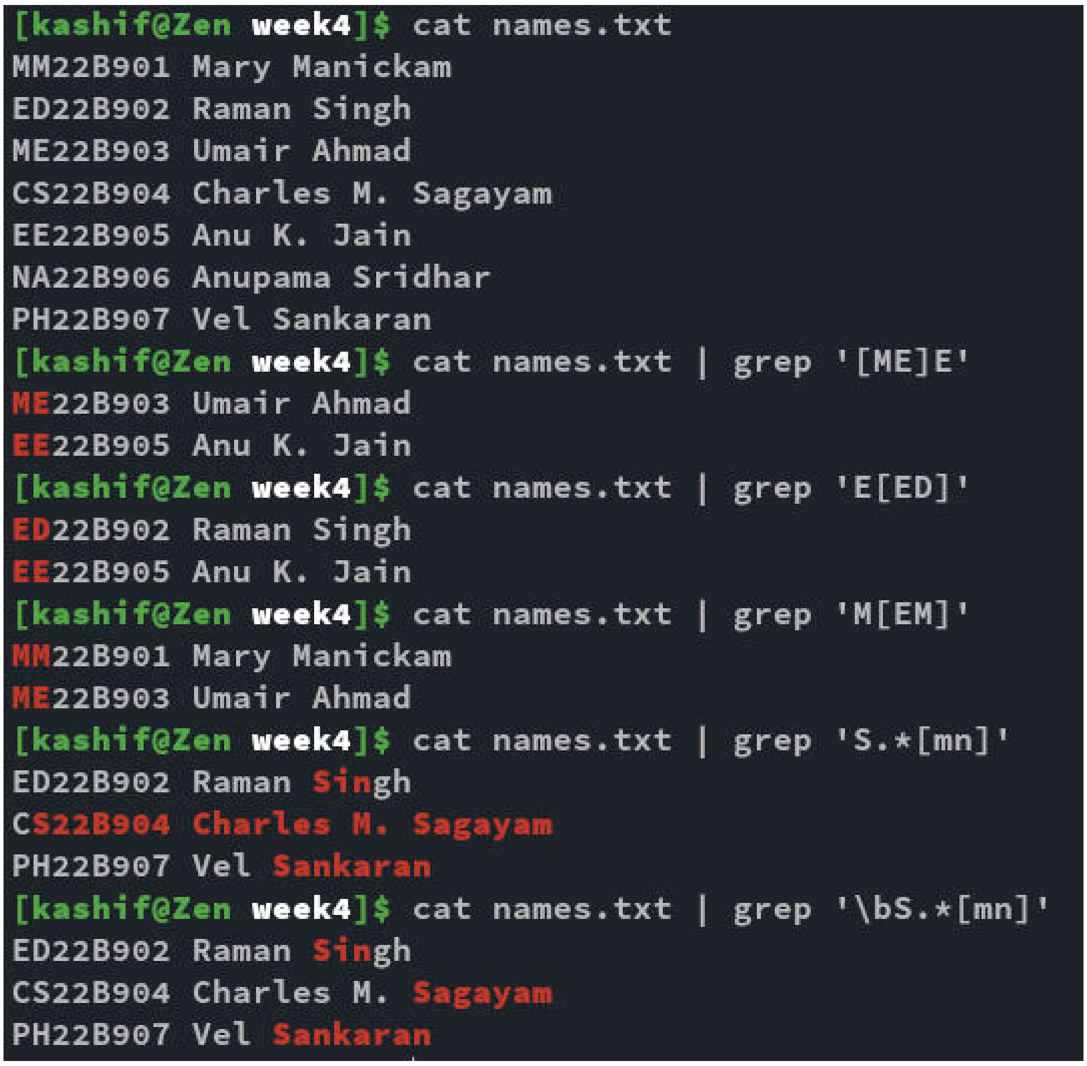
Usage of square brackets `[]`

Here, the first character in the pattern is followed by either of the 2 characters given in `[]`

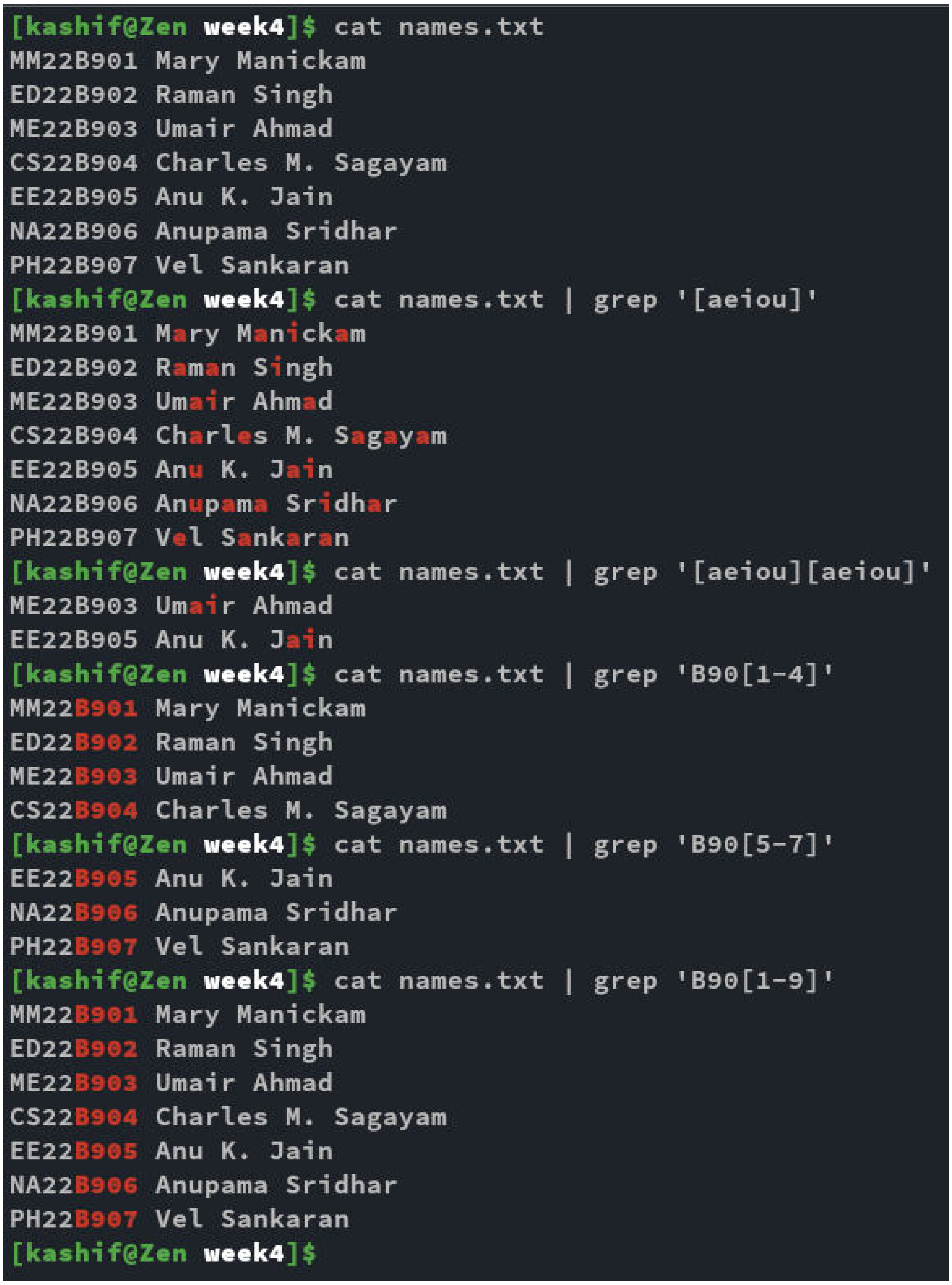
In the `grep 'S.*[mn]'` command, it matches from the start of the line

We add `\b` to mark a word boundary just to match it within a word

> *Had to switch to* `bash` *to show the formatting*

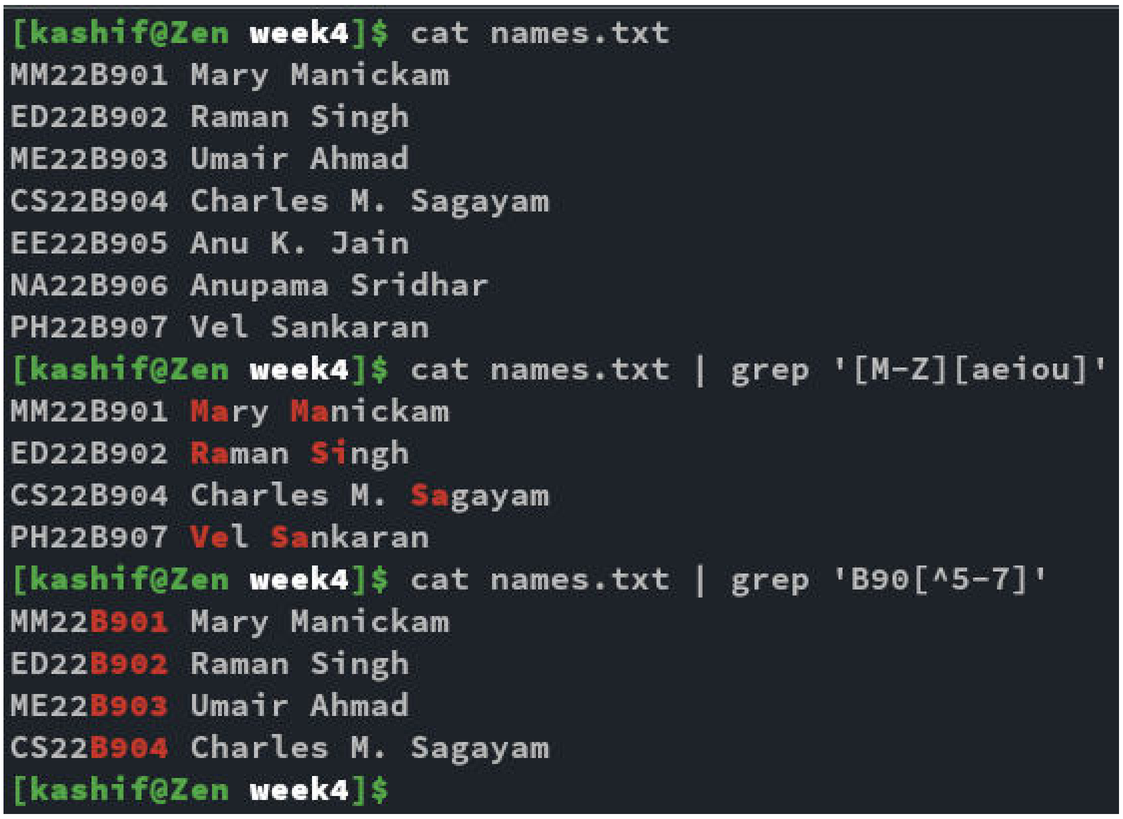


```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '[ME]E'
ME22B903 Umair Ahmad
EE22B905 Anu K. Jain
[kashif@Zen week4]$ cat names.txt | grep 'E[ED]'
ED22B902 Raman Singh
EE22B905 Anu K. Jain
[kashif@Zen week4]$ cat names.txt | grep 'M[EM]'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
[kashif@Zen week4]$ cat names.txt | grep 'S.*[mn]'
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '\bS.*[mn]'
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
PH22B907 Vel Sankaran
```



```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '[aeiou]'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '[aeiou][aeiou]'
ME22B903 Umair Ahmad
EE22B905 Anu K. Jain
[kashif@Zen week4]$ cat names.txt | grep 'B90[1-4]'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
[kashif@Zen week4]$ cat names.txt | grep 'B90[5-7]'
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep 'B90[1-9]'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$
```

The last command in the following screenshot does negation

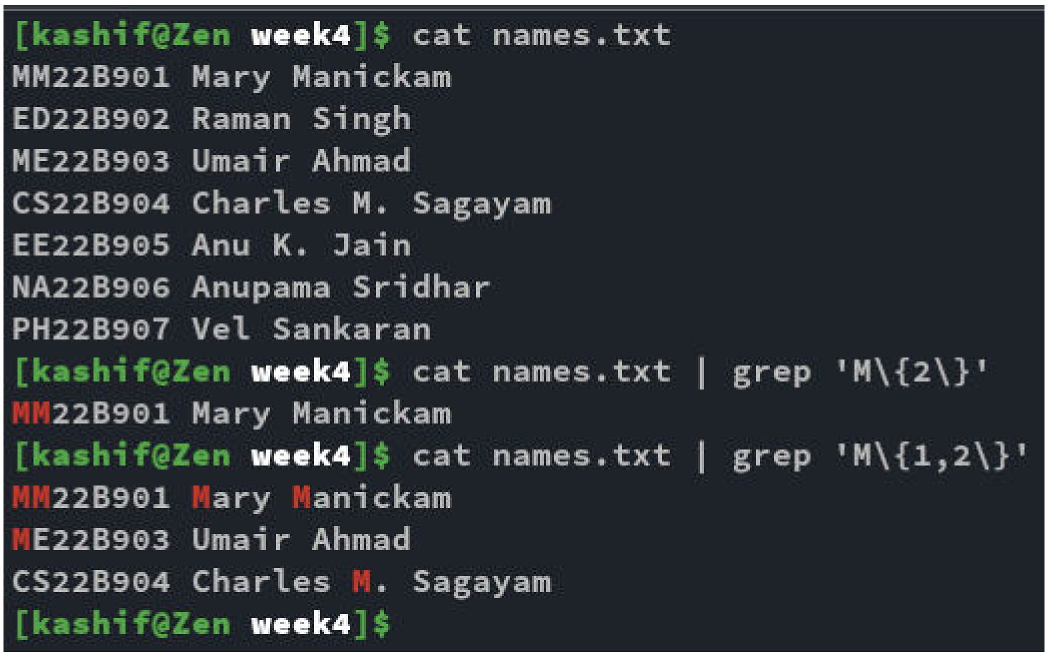


```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '[M-Z][aeiou]'
MM22B901 Mary Manickam
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep 'B90[^5-7]'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
[kashif@Zen week4]$
```

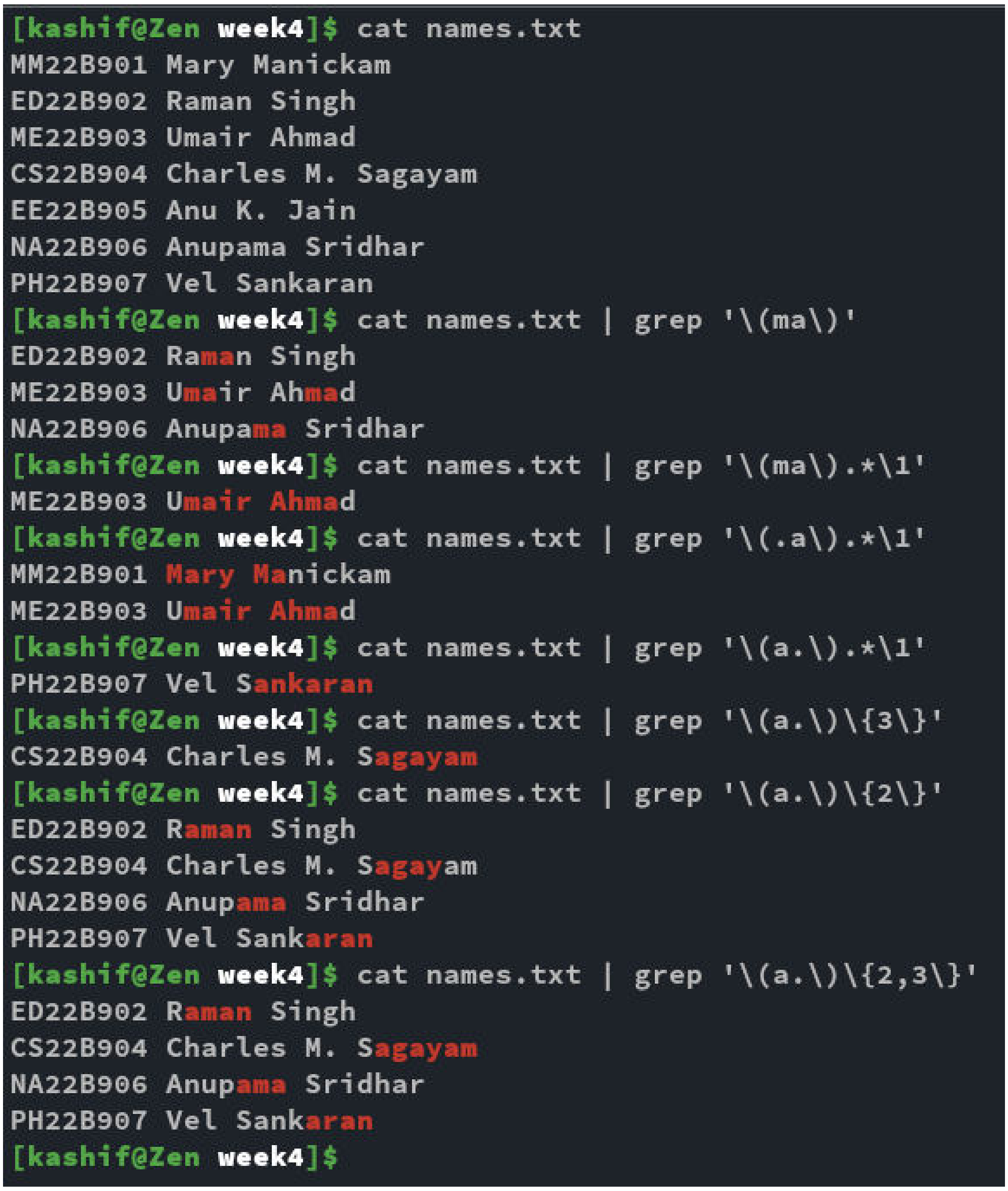
Number of times a character should occur

In the curly braces, we provide the # of times the preceding character should be matched

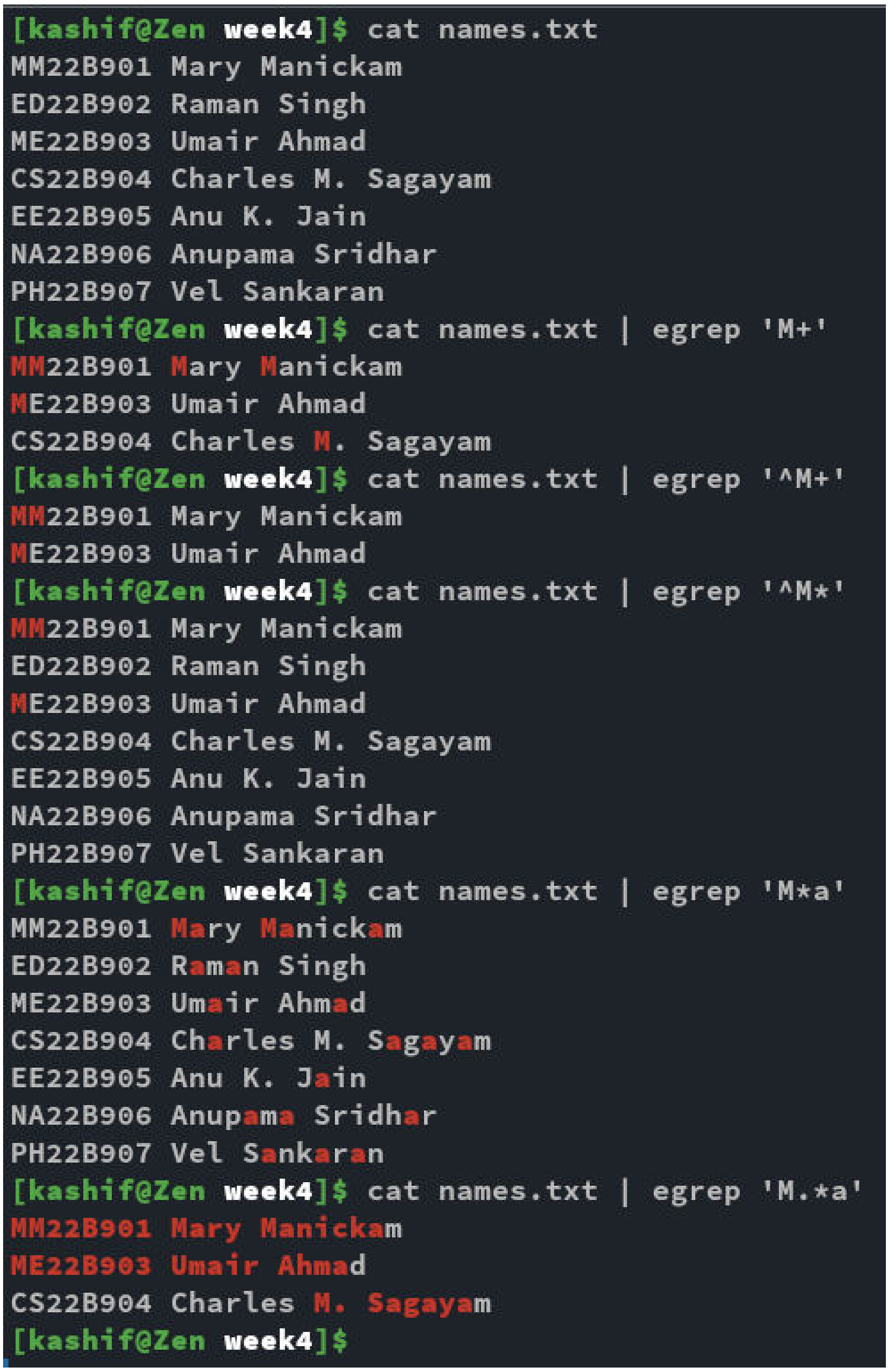
We can pass one number or a multiple numbers separated by comma for their matching



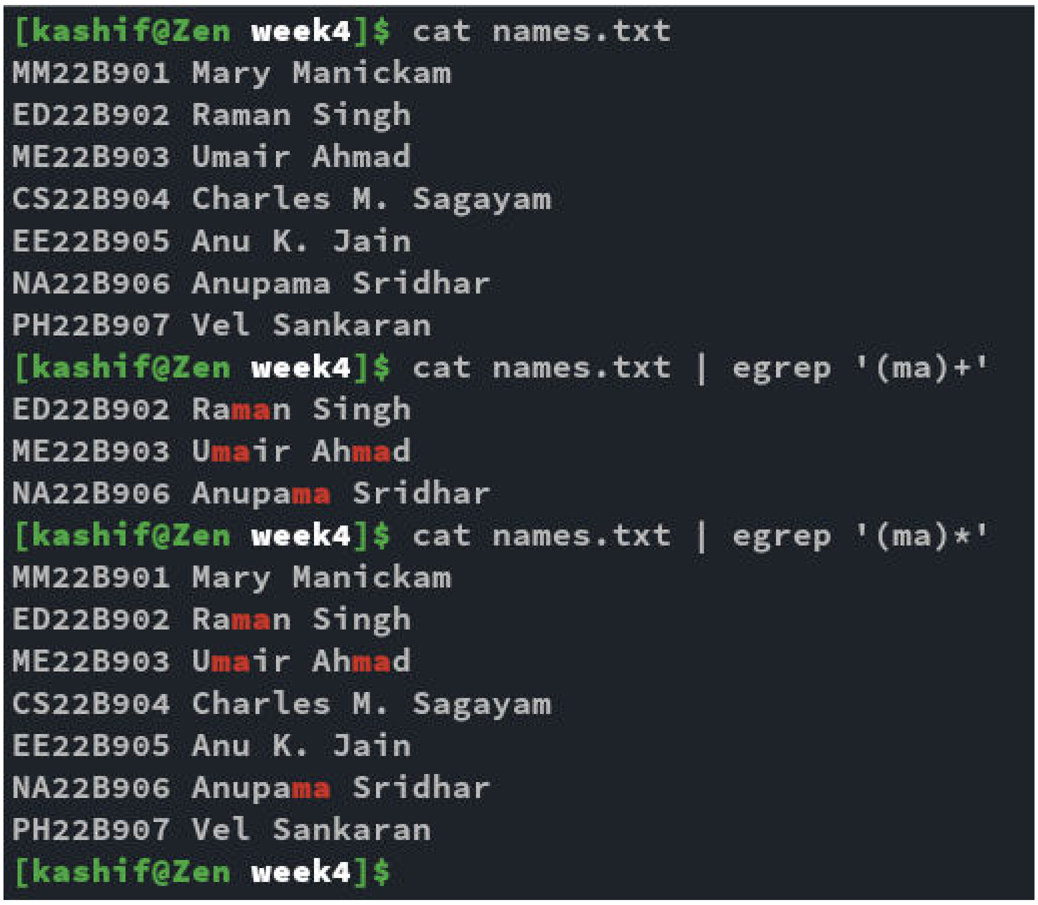
```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep 'M\{2\}'
MM22B901 Mary Manickam
[kashif@Zen week4]$ cat names.txt | grep 'M\{1,2\}'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
[kashif@Zen week4]$
```

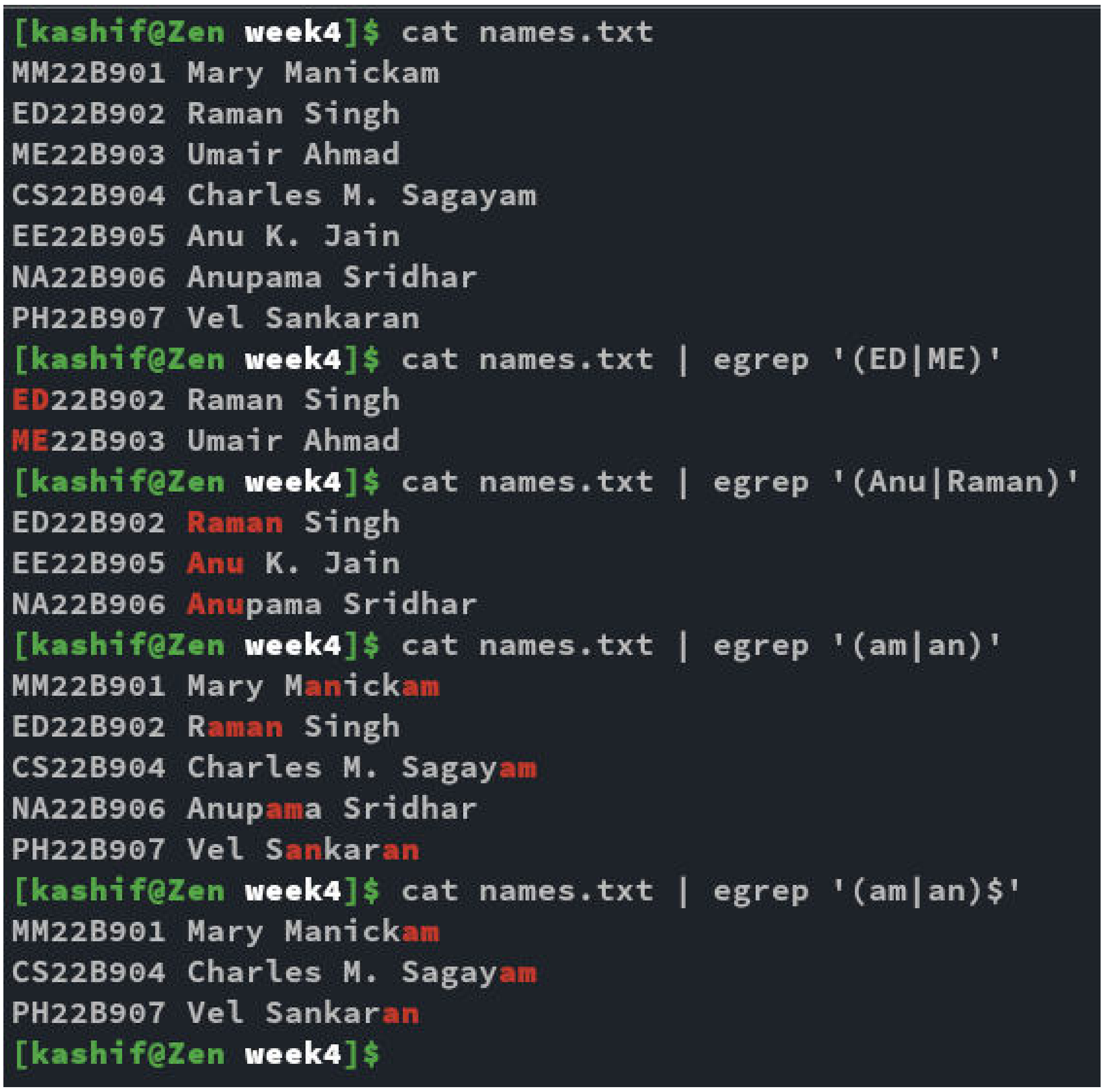


```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '\(ma\)'
ED22B902 Raman Singh
ME22B903 Umair Ahmad
NA22B906 Anupama Sridhar
[kashif@Zen week4]$ cat names.txt | grep '\(ma\).*\1'
ME22B903 Umair Ahmad
[kashif@Zen week4]$ cat names.txt | grep '\(.a\).*\1'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
[kashif@Zen week4]$ cat names.txt | grep '\(a.\).*\1'
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '\(a.\)\{3\}'
CS22B904 Charles M. Sagayam
[kashif@Zen week4]$ cat names.txt | grep '\(a.\)\{2\}'
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | grep '\(a.\)\{2,3\}'
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$
```



```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | egrep 'M+'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
[kashif@Zen week4]$ cat names.txt | egrep '^M+'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
[kashif@Zen week4]$ cat names.txt | egrep '^M*'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | egrep 'M*a'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | egrep 'M.*a'
MM22B901 Mary Manickam
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
[kashif@Zen week4]$
```


```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | egrep '(ma)+'
ED22B902 Raman Singh
ME22B903 Umair Ahmad
NA22B906 Anupama Sridhar
[kashif@Zen week4]$ cat names.txt | egrep '(ma)*'
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$
```



```
[kashif@Zen week4]$ cat names.txt
MM22B901 Mary Manickam
ED22B902 Raman Singh
ME22B903 Umair Ahmad
CS22B904 Charles M. Sagayam
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | egrep '(ED|ME)'
ED22B902 Raman Singh
ME22B903 Umair Ahmad
[kashif@Zen week4]$ cat names.txt | egrep '(Anu|Raman)'
ED22B902 Raman Singh
EE22B905 Anu K. Jain
NA22B906 Anupama Sridhar
[kashif@Zen week4]$ cat names.txt | egrep '(am|an)'
MM22B901 Mary Manickam
ED22B902 Raman Singh
CS22B904 Charles M. Sagayam
NA22B906 Anupama Sridhar
PH22B907 Vel Sankaran
[kashif@Zen week4]$ cat names.txt | egrep '(am|an)$'
MM22B901 Mary Manickam
CS22B904 Charles M. Sagayam
PH22B907 Vel Sankaran
[kashif@Zen week4]$
```

Match package names that are 4 characters long

`dpkg-query -W -f='${Section} ${binary:Package}\n' | egrep ' .{4}$'`

Match package names that are 3 characters long and start with the letter `g`

`dpkg-query -W -f='${Section} ${binary:Package}\n' | egrep ' g.{3}$'`

Match package names that are between 1 to 5 characters long and start with the letter `g`

`dpkg-query -W -f='${Section} ${binary:Package}\n' | egrep ' g.{1,5}$'`

Match package names that are from the `math` category

`dpkg-query -W -f='${Section} ${binary:Package}\n' | egrep '^math'`

make sure to use the `^` (hat) character in the front of the regex pattern to match the `math` category, otherwise it will match package category and the names

Match package names that from KDE

`dpkg-query -W -f='${Section} ${binary:Package}\n' | egrep ' kd.*$'`

To skip empty lines from a file

`cat filename.txt | egrep -v ‘^$’`

- Pick any 12 digit or more number from a text file
    - `egrep ‘[[:digit:]]{12}’ filename.txt`
- Pick any 6 digit or more number from a text file
    - `egrep ‘[[:digit:]]{6}’ filename.txt`

But, there is one problem, if there is any number that is more than 12 digits or more than 6 digits respectively, it will pick that up too

- Pick an exactly 6 digit number from a text file
    - Add a word boundary `\b`
    - `egrep ‘\b[[:digit:]]{6}\b’ filename.txt`
- Pick a roll number (of the type MM22B001) from a text file
    - `egrep ‘\b[[:alpha:]]{2}[[:digit:]]{2}[[:alpha:]][[:digit:]]{3}\b’ filename.txt`
- Pick a URL from a text file (like github.com or https://www.iitm.ac.in)
    - `egrep ‘\b[[:alnum:]]+\.[[:alnum:]]+\b’ filename.txt`

`cut`

A command used to cut lines from files

does horizontal trimming

A sample file `fields.txt`

```
1234;hello world,line-1
234567;welcome cmdline,line-2
3456;parse text,line-3
```

- Cut first 4 characters from the beginning of the lines
    - `cut -c 1-4 fields.txt`
- Cut the next 4 characters from the previous
    - `cut -c 5-8 fields.txt`

- We can skip the beginning or the ending of the substring parameter, *it works like python*
    - Cut 4 chars from the beginning
        - `cut -c -4 fields.txt`
    - Cut from 8th char to the end
        - `cut -c 8- fields.txt`
- Use space as the delimiter and print the first field
    - `cat fields.txt | cut -d “ ” -f 1`
- Similarly, print the second field
    - `cat fields.txt | cut -d “ ” -f 2`
- If we want both fields
    - `cat fields.txt | cut -d “ ” -f 1-2`
- Delimit at a semi-colon `;` and get the first field
    - `cat fields.txt | cut -d “;” -f 1`
- Similarly, get the 2nd field
    - `cat fields.txt | cut -d “;” -f 2`
- We can pipe multiple commands
    - To get the part of the line between `;` and `,`
        - `cat fields.txt | cut -d “;” -f 2 | cut -d “,” -f 1`
    - To do the same thing using `grep` (*similar thing, not exactly the same*)
        - `cat fields.txt | egrep ‘;.*,’`
- To get the part `welcome cmdline` from the file `fields.txt`
    - `cat fields.txt | cut -d “;” -f 2 | cut -d “,” -f 1 | head -n 2 | tail -n 1`