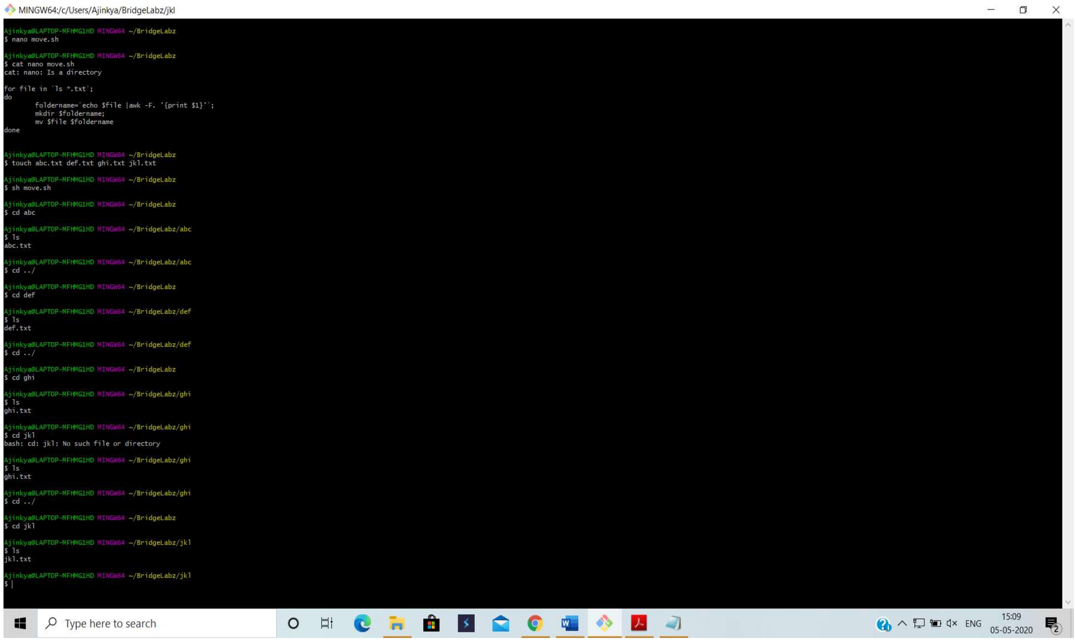
Move files from one folder to the respective folders.

E.g current folder have files abc.txt, def.txt, ghi.txt, jkl.txt
You have to move these files to the folder like abc.txt => abc/ , def.txt => def/ …
Expected outcome -

abc/abc.txt
def/def.txt
ghi/ghi.txt
jkl/jkl.txt

a) Create files in current directory or any temporary directory - abc.txt, def.txt, ghi.txt, jkl.txt
b) Print list of files to move.
c) Segregate basename and extension of a file.
d) Create folder using basename.
e) Move file to newly created folder.
f) Iterate above steps for all files.

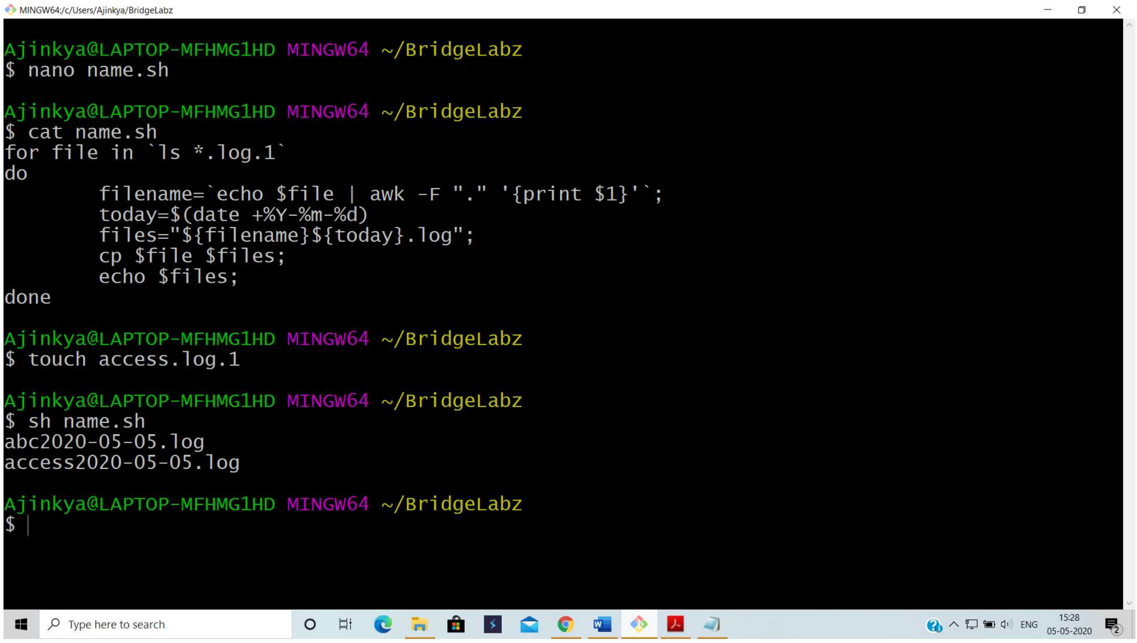


```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ nano move.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cat nano move.sh
cat: nano: Is a directory

for file in `ls *.txt`;
do
        foldername=`echo $file |awk -F. '{print $1}'`;
        mkdir $foldername;
        mv $file $foldername
done


Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ touch abc.txt def.txt ghi.txt jkl.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ sh move.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cd abc

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/abc
$ ls
abc.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/abc
$ cd ../

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cd def

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/def
$ ls
def.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/def
$ cd ../

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cd ghi

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/ghi
$ ls
ghi.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/ghi
$ cd jkl
bash: cd: jkl: No such file or directory

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/ghi
$ ls
ghi.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/ghi
$ cd ../

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cd jkl

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/jkl
$ ls
jkl.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/jkl
$ |
```

Append current date to all log files name which has extension .log.1 from a folder

E.g original file - access.log.1
New updated file name - access-20102019.log

a) Create files with name abc.log.1, def.log.1 , ghi.log.1, jkl.log.1, mno.log.1
b) Print list of files to rename.
c) Segregate basename and extension of a file
d) Print Date Command to show in ddmmyy
e) Append Date to the log file name
f) Iterate above steps for all files which has extension .log.1



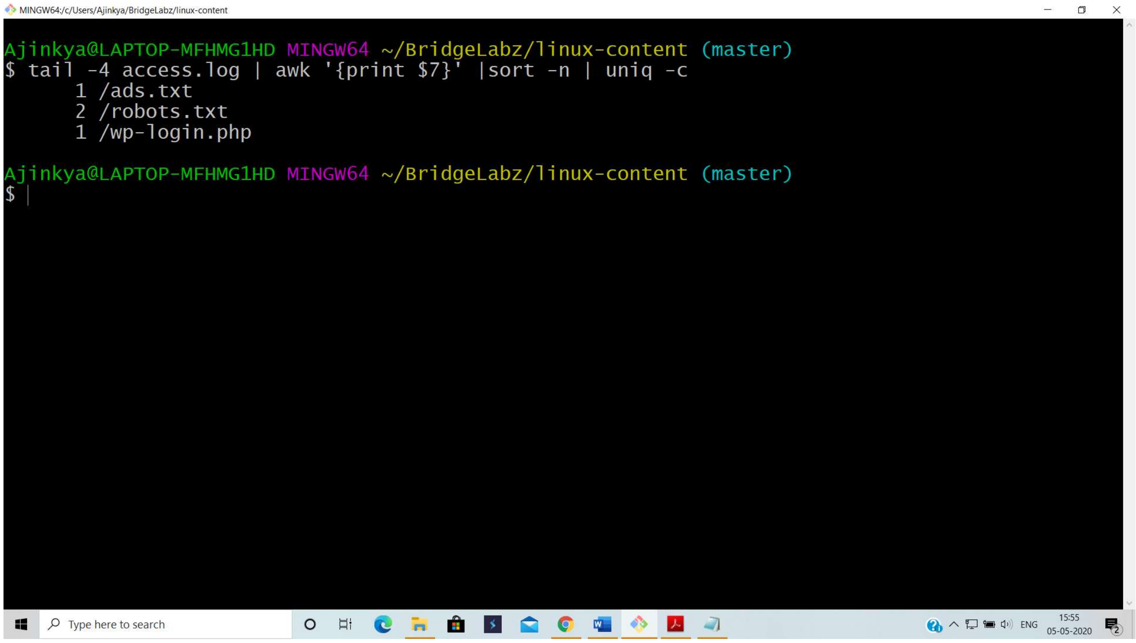
```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ nano name.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cat name.sh
for file in `ls *.log.1`
do
        filename=`echo $file | awk -F "." '{print $1}'`;
        today=$(date +%Y-%m-%d)
        files="${filename}${today}.log";
        cp $file $files;
        echo $files;
done

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ touch access.log.1

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ sh name.sh
abc2020-05-05.log
access2020-05-05.log

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ 
```

Print last 4 frequently access urls count in sorted order from /var/log/httpd/access.log

a) View /var/log/httpd/access.log
b) Print field which has urls data.
c) Sort extracted urls and count it
d) Print 4 unique urls

Expect sample output -

```
3458 /index.html
 300 /api/swagger-ui.html
 100 /favi.ico
  20 /robots.txt
```



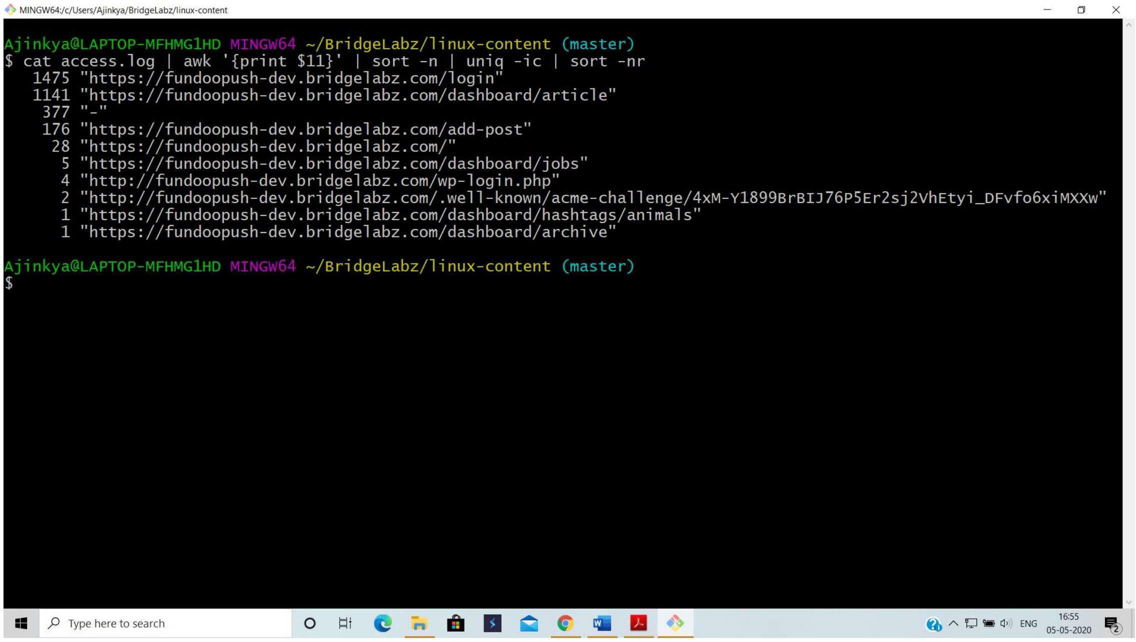
```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ tail -4 access.log | awk '{print $7}' |sort -n | uniq -c
      1 /ads.txt
      2 /robots.txt
      1 /wp-login.php

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ 
```

Print list of last 4 frequently access unique urls at particular hours from /var/log/httpd/access.log

a) View access.log without opening it using editor.
b) Print urls which has given timestamp.
c) Sort extracted urls and count it
d) Print 4 unique urls

Expect sample output -

```
3458 /index.html
 300 /api/swagger-ui.html
 100 /favi.ico
  20 /robots.txt
```



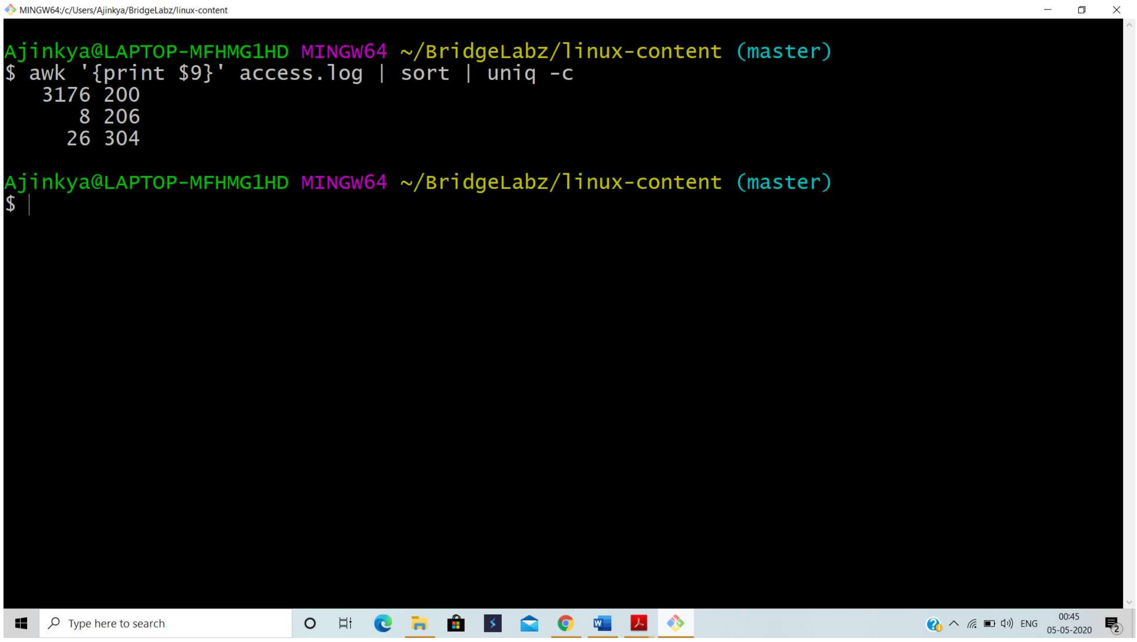
```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat access.log | awk '{print $11}' | sort -n | uniq -ic | sort -nr
   1475 "https://fundoopush-dev.bridgelabz.com/login"
   1141 "https://fundoopush-dev.bridgelabz.com/dashboard/article"
    377 "-"
    176 "https://fundoopush-dev.bridgelabz.com/add-post"
     28 "https://fundoopush-dev.bridgelabz.com/"
      5 "https://fundoopush-dev.bridgelabz.com/dashboard/jobs"
      4 "http://fundoopush-dev.bridgelabz.com/wp-login.php"
      2 "http://fundoopush-dev.bridgelabz.com/.well-known/acme-challenge/4xM-Y1899BrBIJ76P5Er2sj2VhEtyi_DFvfo6xiMXXw"
      1 "https://fundoopush-dev.bridgelabz.com/dashboard/hashtags/animals"
      1 "https://fundoopush-dev.bridgelabz.com/dashboard/archive"

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$
```

Print list of web response code count in the unique sorted order at specific hours

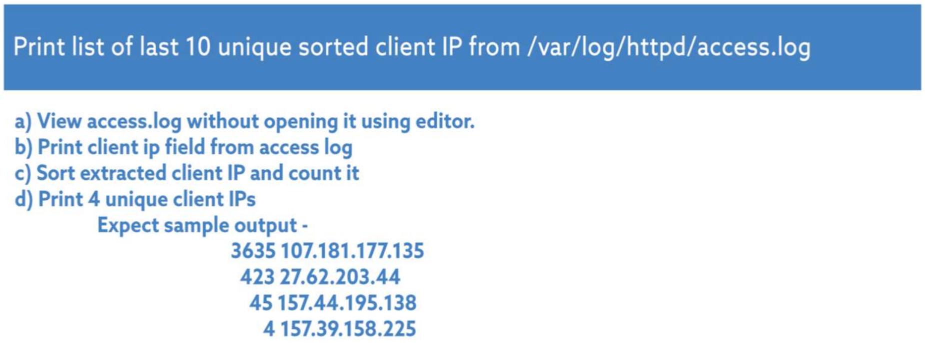
a) View access.log without opening it using editor.
b) Print web response code field which has given timestamp
c) Sort extracted response code and count it
d) Print 4 unique response code count

Expected sample output -

```
1000 200
 100 304
```



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ awk '{print $9}' access.log | sort | uniq -c
   3176 200
      8 206
     26 304

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$
```

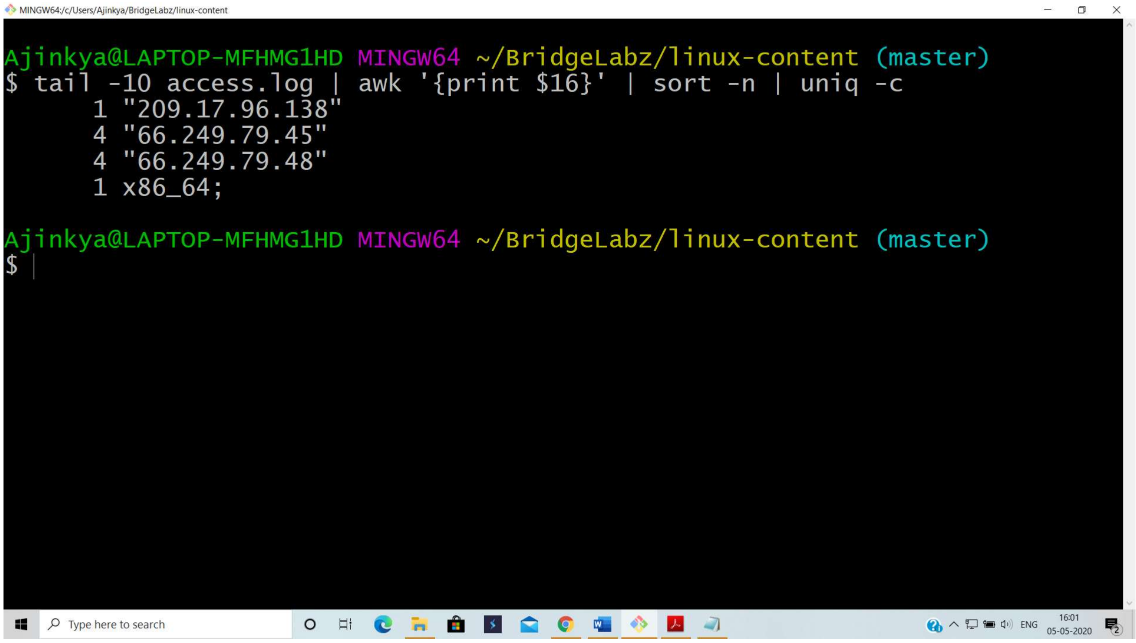


Print list of last 10 unique sorted client IP from /var/log/httpd/access.log

a) View access.log without opening it using editor.
b) Print client ip field from access log
c) Sort extracted client IP and count it
d) Print 4 unique client IPs

Expect sample output -

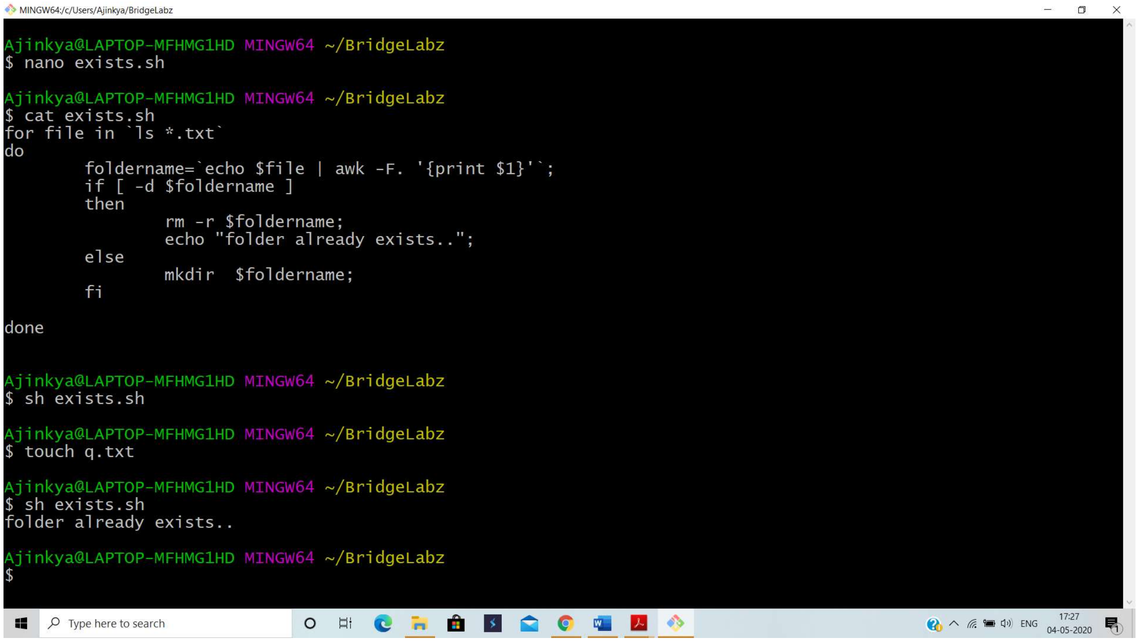
```
3635 107.181.177.135
 423 27.62.203.44
  45 157.44.195.138
   4 157.39.158.225
```



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ tail -10 access.log | awk '{print $16}' | sort -n | uniq -c
      1 "209.17.96.138"
      4 "66.249.79.45"
      4 "66.249.79.48"
      1 x86_64;

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$
```

Check if a folder exists or not. If it’s not present, create it

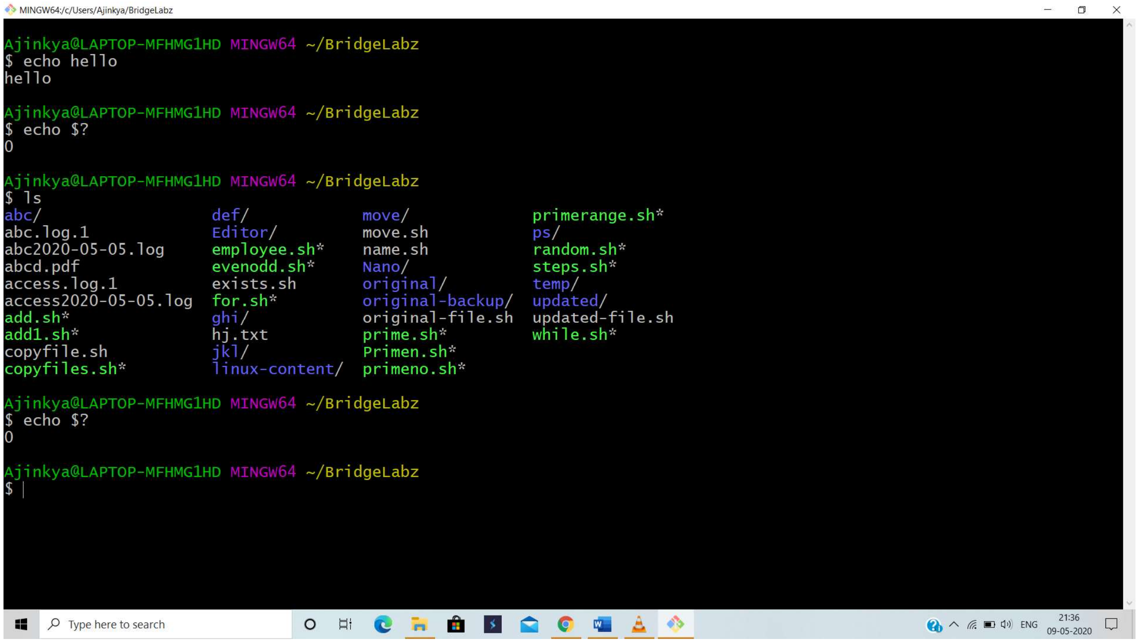
a) Test if particular folder exists in current directory or not
b) If its doesn’t exists then create it else print “folder already exists..”



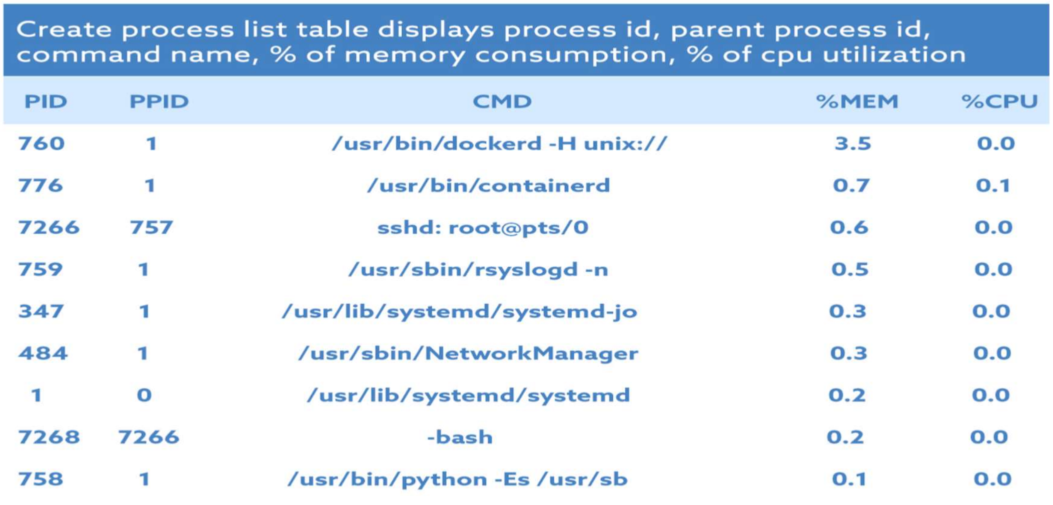
```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ nano exists.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cat exists.sh
for file in `ls *.txt`
do
        foldername=`echo $file | awk -F. '{print $1}'`;
        if [ -d $foldername ]
        then
                rm -r $foldername;
                echo "folder already exists..";
        else
                mkdir  $foldername;
        fi

done


Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ sh exists.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ touch q.txt

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ sh exists.sh
folder already exists..

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$
```

Execute command “hello” and “ls” and check its execution status and print whether command executed successful or not.

a) Execute “hello” command at command prompt
b) Check execution status of “hello” command
c) Execute “ls” command at command prompt
d) Check execution status of “ls” command

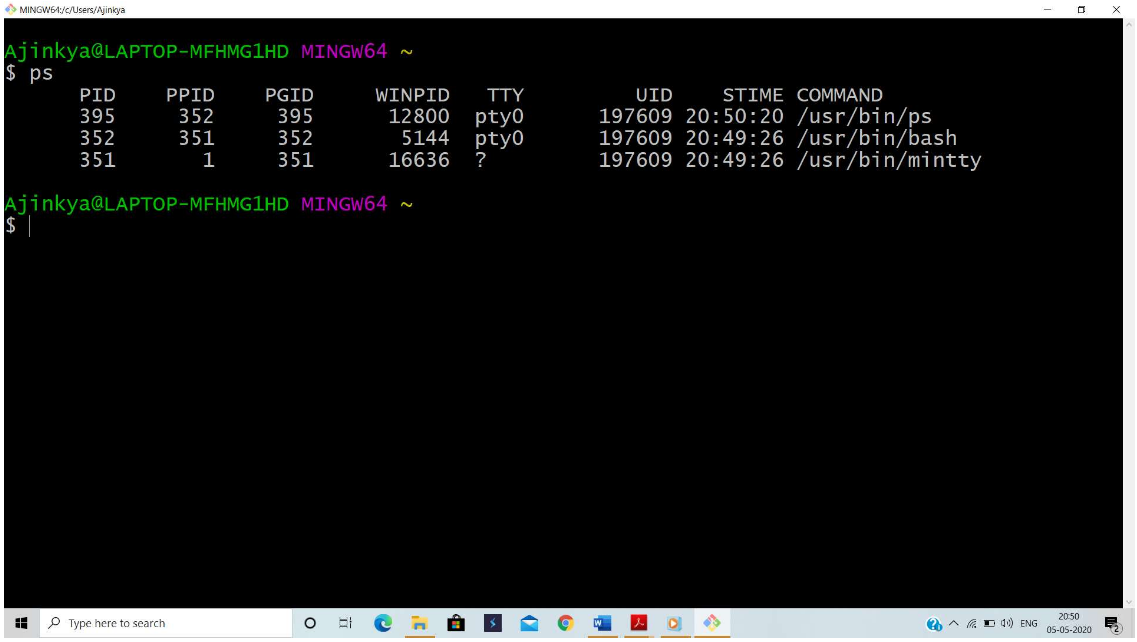


```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo hello
hello

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $?
0

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ ls
abc/                  def/            move/             primerange.sh*
abc.log.1             Editor/         move.sh           ps/
abc2020-05-05.log     employee.sh*    name.sh           random.sh*
abcd.pdf              evenodd.sh*     Nano/             steps.sh*
access.log.1          exists.sh       original/         temp/
access2020-05-05.log  for.sh*         original-backup/  updated/
add.sh*               ghi/            original-file.sh  updated-file.sh
add1.sh*              hj.txt          prime.sh*         while.sh*
copyfile.sh           jkl/            Primen.sh*
copyfiles.sh*         linux-content/  primeno.sh*

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $?
0

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ 
```



Create process list table displays process id, parent process id, command name, % of memory consumption, % of cpu utilization

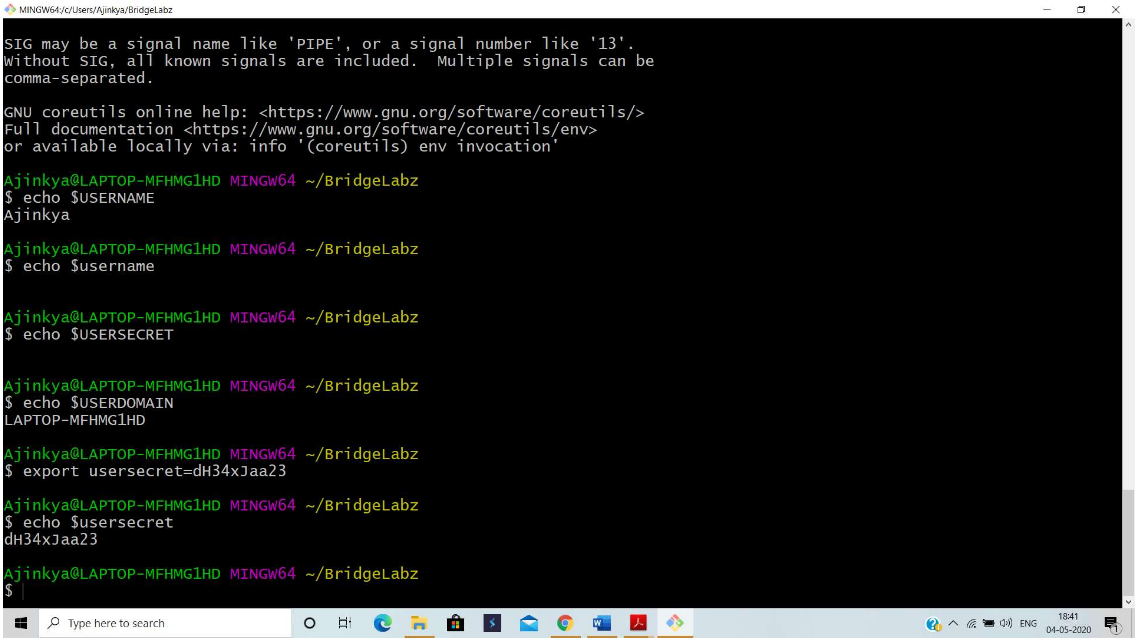
PID	PPID	CMD	%MEM	%CPU
760	1	/usr/bin/dockerd -H unix://	3.5	0.0
776	1	/usr/bin/containerd	0.7	0.1
7266	757	sshd: root@pts/0	0.6	0.0
759	1	/usr/sbin/rsyslogd -n	0.5	0.0
347	1	/usr/lib/systemd/systemd-jo	0.3	0.0
484	1	/usr/sbin/NetworkManager	0.3	0.0
1	0	/usr/lib/systemd/systemd	0.2	0.0
7268	7266	-bash	0.2	0.0
758	1	/usr/bin/python -Es /usr/sb	0.1	0.0



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~
$ ps
      PID    PPID    PGID     WINPID   TTY         UID    STIME COMMAND
      395     352     395      12800  pty0      197609 20:50:20 /usr/bin/ps
      352     351     352       5144  pty0      197609 20:49:26 /usr/bin/bash
      351       1     351      16636  ?         197609 20:49:26 /usr/bin/mintty

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~
$ 
```

Set environment usersecret="dH34xJaa23" if its already not set

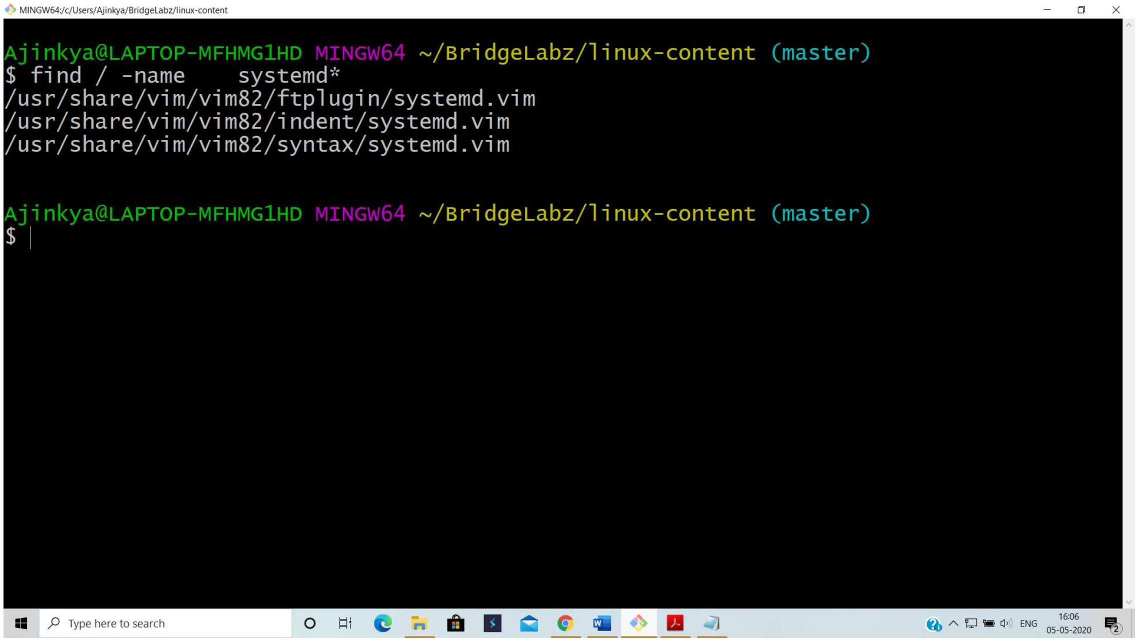
a) Check whether environment variable usersecret assigned any value or not
b) Print error if usersecret already set
c) Set environment variable usersecret to given value.



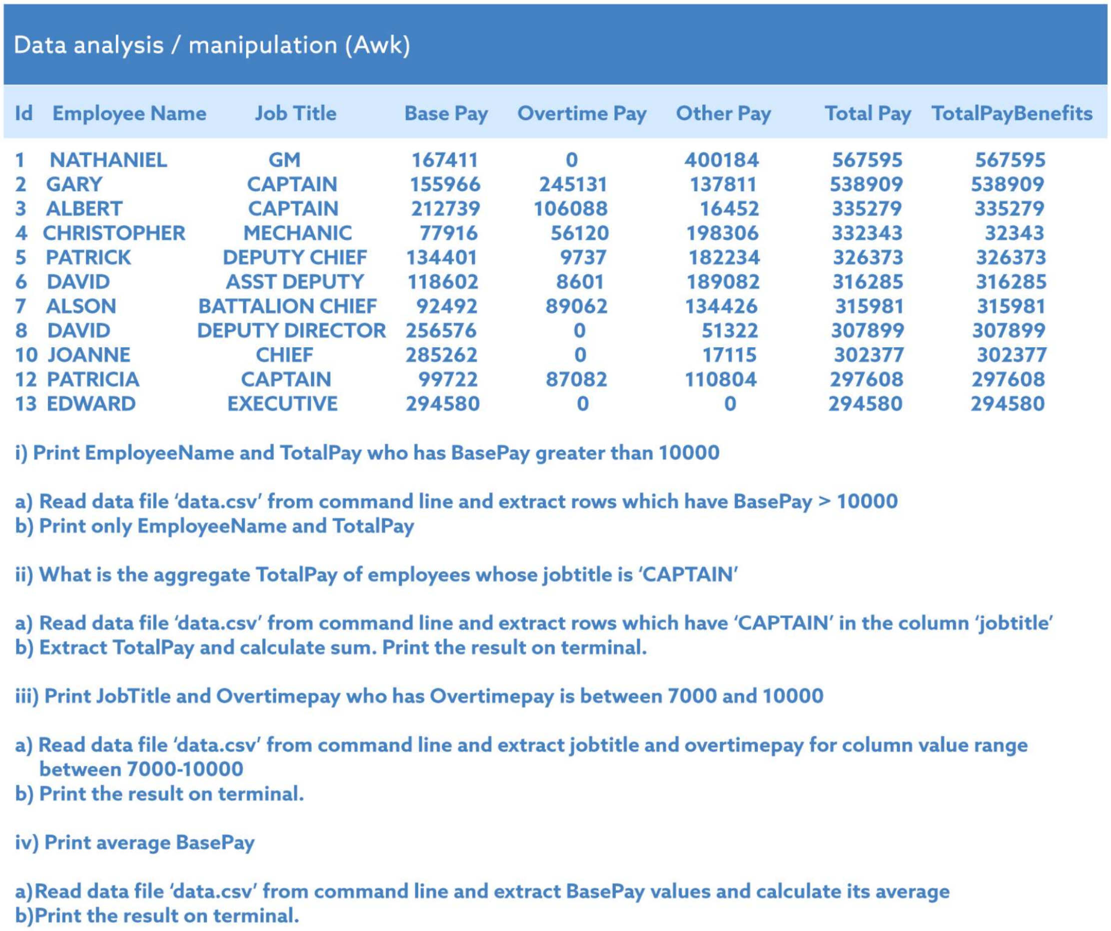
```
SIG may be a signal name like 'PIPE', or a signal number like '13'.
Without SIG, all known signals are included.  Multiple signals can be
comma-separated.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/env>
or available locally via: info '(coreutils) env invocation'

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $USERNAME
Ajinkya

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $username


Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $USERSECRET


Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $USERDOMAIN
LAPTOP-MFHMG1HD

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ export usersecret=dH34xJaa23

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ echo $usersecret
dH34xJaa23

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ |
```

Find a word "systemd" from all log files in the folder /var/log and print number of occurrence more than 0 against each file.

a) Use linux command to search word and print occurrence



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ find / -name    systemd*
/usr/share/vim/vim82/ftplugin/systemd.vim
/usr/share/vim/vim82/indent/systemd.vim
/usr/share/vim/vim82/syntax/systemd.vim


Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$
```



Data analysis / manipulation (Awk)

Id	Employee Name	Job Title	Base Pay	Overtime Pay	Other Pay	Total Pay	TotalPayBenefits
1	NATHANIEL	GM	167411	0	400184	567595	567595
2	GARY	CAPTAIN	155966	245131	137811	538909	538909
3	ALBERT	CAPTAIN	212739	106088	16452	335279	335279
4	CHRISTOPHER	MECHANIC	77916	56120	198306	332343	32343
5	PATRICK	DEPUTY CHIEF	134401	9737	182234	326373	326373
6	DAVID	ASST DEPUTY	118602	8601	189082	316285	316285
7	ALSON	BATTALION CHIEF	92492	89062	134426	315981	315981
8	DAVID	DEPUTY DIRECTOR	256576	0	51322	307899	307899
10	JOANNE	CHIEF	285262	0	17115	302377	302377
12	PATRICIA	CAPTAIN	99722	87082	110804	297608	297608
13	EDWARD	EXECUTIVE	294580	0	0	294580	294580

i) Print EmployeeName and TotalPay who has BasePay greater than 10000

a) Read data file 'data.csv' from command line and extract rows which have BasePay > 10000
b) Print only EmployeeName and TotalPay

ii) What is the aggregate TotalPay of employees whose jobtitle is 'CAPTAIN'

a) Read data file 'data.csv' from command line and extract rows which have 'CAPTAIN' in the column 'jobtitle'
b) Extract TotalPay and calculate sum. Print the result on terminal.

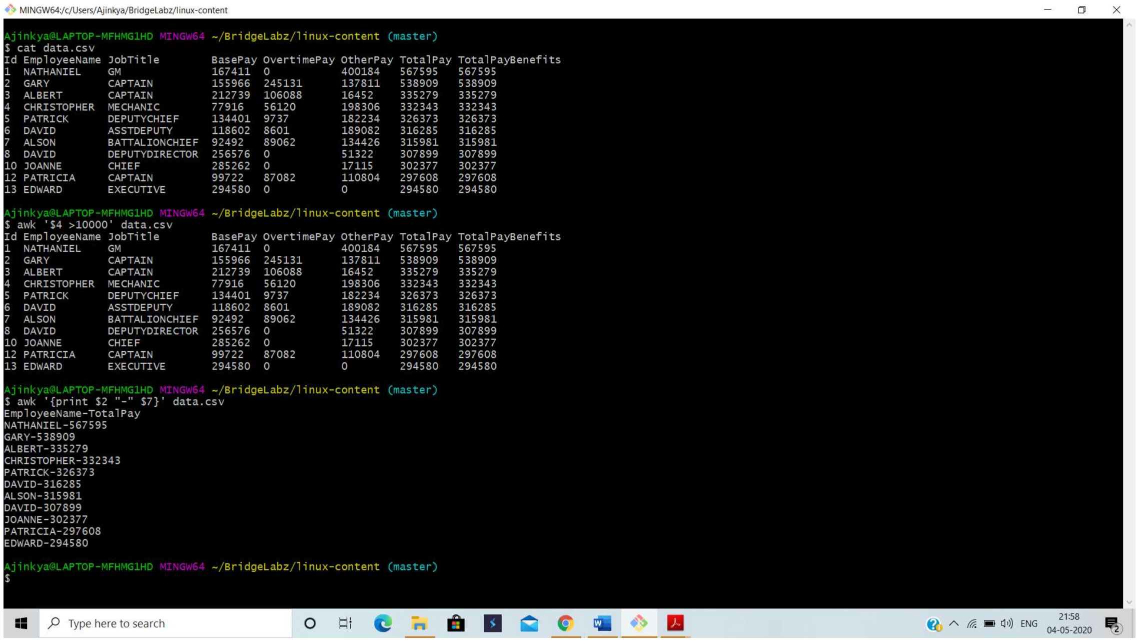
iii) Print JobTitle and Overtimepay who has Overtimepay is between 7000 and 10000

a) Read data file 'data.csv' from command line and extract jobtitle and overtimepay for column value range between 7000-10000
b) Print the result on terminal.

iv) Print average BasePay

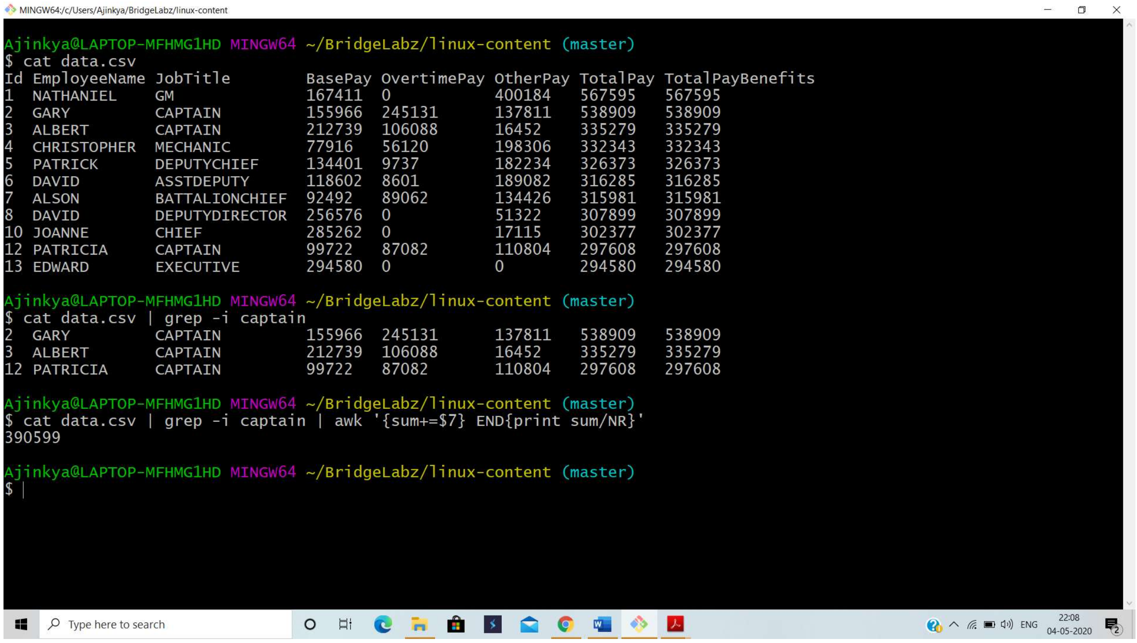
a)Read data file 'data.csv' from command line and extract BasePay values and calculate its average
b)Print the result on terminal.

i)



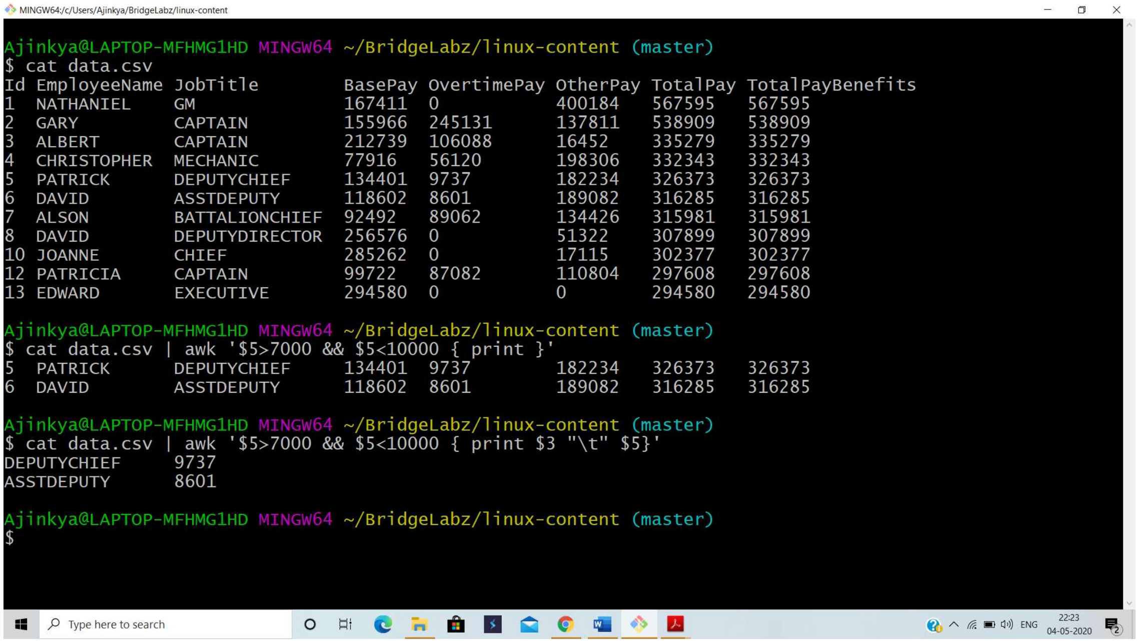
```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv
Id EmployeeName JobTitle        BasePay OvertimePay OtherPay TotalPay TotalPayBenefits
1  NATHANIEL    GM              167411  0           400184   567595   567595
2  GARY         CAPTAIN         155966  245131      137811   538909   538909
3  ALBERT       CAPTAIN         212739  106088      16452    335279   335279
4  CHRISTOPHER  MECHANIC        77916   56120       198306   332343   332343
5  PATRICK      DEPUTYCHIEF     134401  9737        182234   326373   326373
6  DAVID        ASSTDEPUTY      118602  8601        189082   316285   316285
7  ALSON        BATTALIONCHIEF  92492   89062       134426   315981   315981
8  DAVID        DEPUTYDIRECTOR  256576  0           51322    307899   307899
10 JOANNE       CHIEF           285262  0           17115    302377   302377
12 PATRICIA     CAPTAIN         99722   87082       110804   297608   297608
13 EDWARD       EXECUTIVE       294580  0           0        294580   294580

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ awk '$4 >10000' data.csv
Id EmployeeName JobTitle        BasePay OvertimePay OtherPay TotalPay TotalPayBenefits
1  NATHANIEL    GM              167411  0           400184   567595   567595
2  GARY         CAPTAIN         155966  245131      137811   538909   538909
3  ALBERT       CAPTAIN         212739  106088      16452    335279   335279
4  CHRISTOPHER  MECHANIC        77916   56120       198306   332343   332343
5  PATRICK      DEPUTYCHIEF     134401  9737        182234   326373   326373
6  DAVID        ASSTDEPUTY      118602  8601        189082   316285   316285
7  ALSON        BATTALIONCHIEF  92492   89062       134426   315981   315981
8  DAVID        DEPUTYDIRECTOR  256576  0           51322    307899   307899
10 JOANNE       CHIEF           285262  0           17115    302377   302377
12 PATRICIA     CAPTAIN         99722   87082       110804   297608   297608
13 EDWARD       EXECUTIVE       294580  0           0        294580   294580

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ awk '{print $2 "-" $7}' data.csv
EmployeeName-TotalPay
NATHANIEL-567595
GARY-538909
ALBERT-335279
CHRISTOPHER-332343
PATRICK-326373
DAVID-316285
ALSON-315981
DAVID-307899
JOANNE-302377
PATRICIA-297608
EDWARD-294580

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$
```

ii)



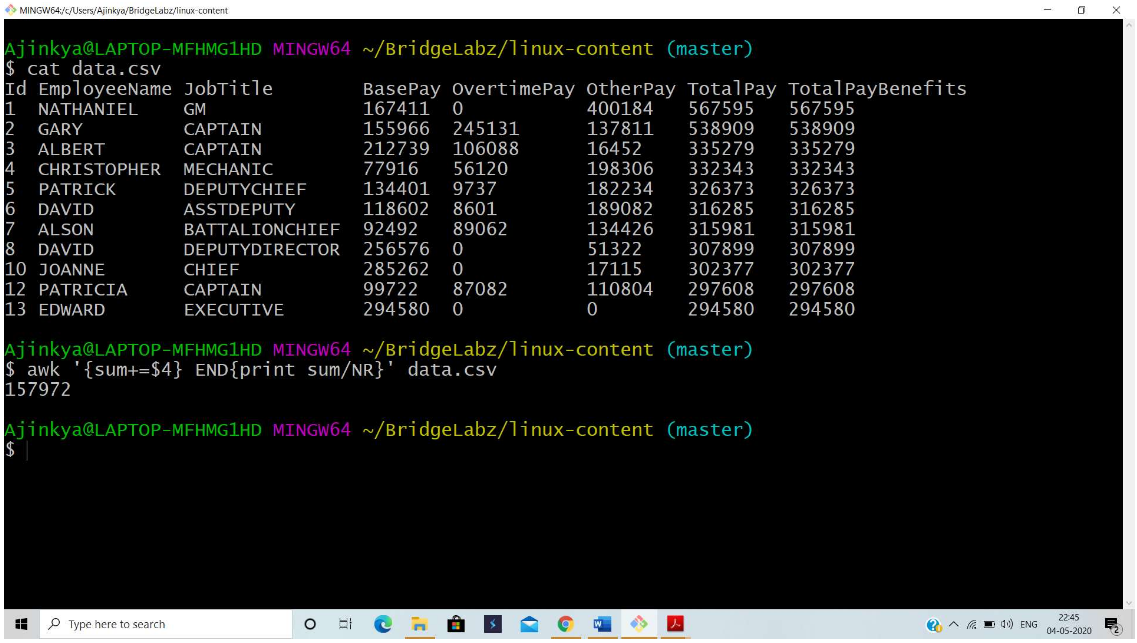
```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv
Id EmployeeName JobTitle        BasePay OvertimePay OtherPay TotalPay TotalPayBenefits
1  NATHANIEL    GM              167411  0           400184   567595   567595
2  GARY         CAPTAIN         155966  245131      137811   538909   538909
3  ALBERT       CAPTAIN         212739  106088      16452    335279   335279
4  CHRISTOPHER  MECHANIC        77916   56120       198306   332343   332343
5  PATRICK      DEPUTYCHIEF     134401  9737        182234   326373   326373
6  DAVID        ASSTDEPUTY      118602  8601        189082   316285   316285
7  ALSON        BATTALIONCHIEF  92492   89062       134426   315981   315981
8  DAVID        DEPUTYDIRECTOR  256576  0           51322    307899   307899
10 JOANNE       CHIEF           285262  0           17115    302377   302377
12 PATRICIA     CAPTAIN         99722   87082       110804   297608   297608
13 EDWARD       EXECUTIVE       294580  0           0        294580   294580

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv | grep -i captain
2  GARY         CAPTAIN         155966  245131      137811   538909   538909
3  ALBERT       CAPTAIN         212739  106088      16452    335279   335279
12 PATRICIA     CAPTAIN         99722   87082       110804   297608   297608

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv | grep -i captain | awk '{sum+=$7} END{print sum/NR}'
390599

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ |
```

iii)



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv
Id EmployeeName JobTitle        BasePay OvertimePay OtherPay TotalPay TotalPayBenefits
1  NATHANIEL    GM              167411  0           400184   567595   567595
2  GARY         CAPTAIN         155966  245131      137811   538909   538909
3  ALBERT       CAPTAIN         212739  106088      16452    335279   335279
4  CHRISTOPHER  MECHANIC        77916   56120       198306   332343   332343
5  PATRICK      DEPUTYCHIEF     134401  9737        182234   326373   326373
6  DAVID        ASSTDEPUTY      118602  8601        189082   316285   316285
7  ALSON        BATTALIONCHIEF  92492   89062       134426   315981   315981
8  DAVID        DEPUTYDIRECTOR  256576  0           51322    307899   307899
10 JOANNE       CHIEF           285262  0           17115    302377   302377
12 PATRICIA     CAPTAIN         99722   87082       110804   297608   297608
13 EDWARD       EXECUTIVE       294580  0           0        294580   294580

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv | awk '$5>7000 && $5<10000 { print }'
5  PATRICK      DEPUTYCHIEF     134401  9737        182234   326373   326373
6  DAVID        ASSTDEPUTY      118602  8601        189082   316285   316285

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv | awk '$5>7000 && $5<10000 { print $3 "\t" $5}'
DEPUTYCHIEF     9737
ASSTDEPUTY      8601

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$
```

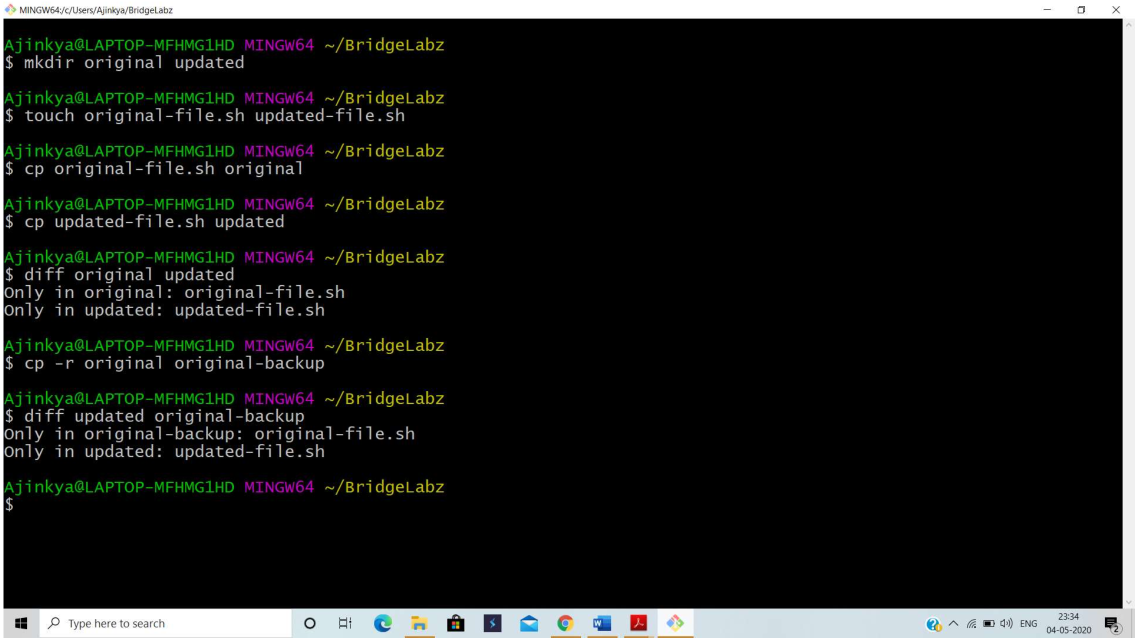
iv)



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ cat data.csv
Id EmployeeName JobTitle        BasePay OvertimePay OtherPay TotalPay TotalPayBenefits
1  NATHANIEL    GM              167411  0           400184   567595   567595
2  GARY         CAPTAIN         155966  245131      137811   538909   538909
3  ALBERT       CAPTAIN         212739  106088      16452    335279   335279
4  CHRISTOPHER  MECHANIC        77916   56120       198306   332343   332343
5  PATRICK      DEPUTYCHIEF     134401  9737        182234   326373   326373
6  DAVID        ASSTDEPUTY      118602  8601        189082   316285   316285
7  ALSON        BATTALIONCHIEF  92492   89062       134426   315981   315981
8  DAVID        DEPUTYDIRECTOR  256576  0           51322    307899   307899
10 JOANNE       CHIEF           285262  0           17115    302377   302377
12 PATRICIA     CAPTAIN         99722   87082       110804   297608   297608
13 EDWARD       EXECUTIVE       294580  0           0        294580   294580

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ awk '{sum+=$4} END{print sum/NR}' data.csv
157972

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz/linux-content (master)
$ |
```

Find the difference between original file and the updated file.
Apply changes to the original file.

a) Create two directories as “original’ and “updated”
b) Copy given file ‘original-file.sh’ to the folder “original” and “updated-file.sh” to the folder “updated”
c) Find the difference between these directories using linux command
d) Make copy of folder “original” to some other directory as “original-backup” and apply changes to ‘original-file.sh’ file
e) Verify that both folders “updated” and “original-backup” have no difference.



```
Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ mkdir original updated

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ touch original-file.sh updated-file.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cp original-file.sh original

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cp updated-file.sh updated

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ diff original updated
Only in original: original-file.sh
Only in updated: updated-file.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ cp -r original original-backup

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$ diff updated original-backup
Only in original-backup: original-file.sh
Only in updated: updated-file.sh

Ajinkya@LAPTOP-MFHMG1HD MINGW64 ~/BridgeLabz
$
```