

A) Write a shell script that determines the period for which a specified user is working on the system.

- echo "Enter the desired username: "
- read a
- last Sa

```
Terminal Nov 20 02:01
sajid@Ubuntum1: ~/Desktop
sajid@Ubuntum1:~/Desktop$ nano sj.sh
sajid@Ubuntum1:~/Desktop$ ./sj.sh
Enter the desired username
sajid
sajid      tty2      tty2      Sun Nov 20 01:07   gone - no logout
sajid      tty2      tty2      Fri Nov  4 09:08 - crash (15+22:28)
sajid      tty2      tty2      Wed Nov  2 05:58 - 06:31 (00:33)
sajid      tty2      tty2      Wed Oct 12 05:57 - crash (21+00:01)
sajid      tty2      tty2      Fri Sep 30 09:28 - down (00:59)
sajid      tty2      tty2      Fri Sep 30 09:18 - crash (00:08)
sajid      tty2      tty2      Fri Sep 30 09:09 - crash (00:08)
sajid      tty2      tty2      Wed Sep 28 07:16 - down (00:00)
sajid      tty2      tty2      Wed Sep 28 06:47 - crash (00:28)
sajid      tty2      tty2      Wed Sep 28 06:41 - crash (00:05)
sajid      tty2      tty2      Wed Sep 28 05:58 - 06:41 (00:43)
sajid      tty2      tty2      Wed Sep 21 05:51 - crash (7+00:04)
sajid      tty2      tty2      Fri Sep 16 03:46 - 04:33 (00:47)
sajid      tty2      tty2      Thu Sep 15 09:08 - 09:12 (00:03)
sajid      tty2      tty2      Wed Sep 14 01:41 - down (00:01)
sajid      tty2      tty2      Wed Sep 14 01:04 - down (00:33)
```

B) Write a shell script that displays all the lines between start and end line numbers passed as argument.

- echo "Enter filename:"
- read a
- echo "Enter the starting line:"
- read sl
- echo "Enter the ending line"
- read el
- sed -n \$sl,\$el\p \$a

```
sajid@Ubuntum1: ~/Desktop
GNU nano 6.2 sn.sh *
echo "Enter the filename: "
read a
echo "Enter the starting line: "
read sl
echo "Enter the ending line"
read el
sed -n $sl,$el\p $a
```

```
sajid@Ubuntum1:~/Desktop$ nano sn.sh
sajid@Ubuntum1:~/Desktop$ ./sn.sh
Enter the filename:
file.txt
Enter the starting line:
1
Enter the ending line
5
Everything you
ever wanted
is on the
other side
sajid@Ubuntum1:~/Desktop$
```

C) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

- if [\$# -eq 0]
- then
- echo NO ARGUMENT
- else
- pattern=\$1
- shift

- for fname in \$*
- do
- if [-f \$fname]
- then
- echo DELETING: \$pattern FROM : \$fname
- sed '/'\$pattern'/d' \$fname
- else
- echo \$fname : FILENAME NOT FOUND
- fi
- done
- fi

```
sajid@Ubuntum1: ~/Desktop
GNU nano 6.2 ns.sh
if [ $# -eq 0 ]
then
echo NO ARGUMENTS
else
pattern=$1
shift
for fname in $*
do
if [ -f $fname ]
then
echo DELETING:$pattern FROM:$fname
sed '/'$pattern'/d' $fname
else
echo $fname : FILENAME NOT FOUND
fi
done
fi
```

```
Nov 20 03:05
sajid@Ubuntum1: ~/Desktop
sajid@Ubuntum1:~/Desktop$ nano ns.sh
sajid@Ubuntum1:~/Desktop$ cat file.txt
Everything you
ever wanted
is on the
other side
of fear
sajid@Ubuntum1:~/Desktop$ ./ns.sh
NO ARGUMENTS
sajid@Ubuntum1:~/Desktop$ ./ns.sh wanted file.txt
DELETING:wanted FROM:file.txt
Everything you
is on the
other side
of fear
sajid@Ubuntum1:~/Desktop$
```

