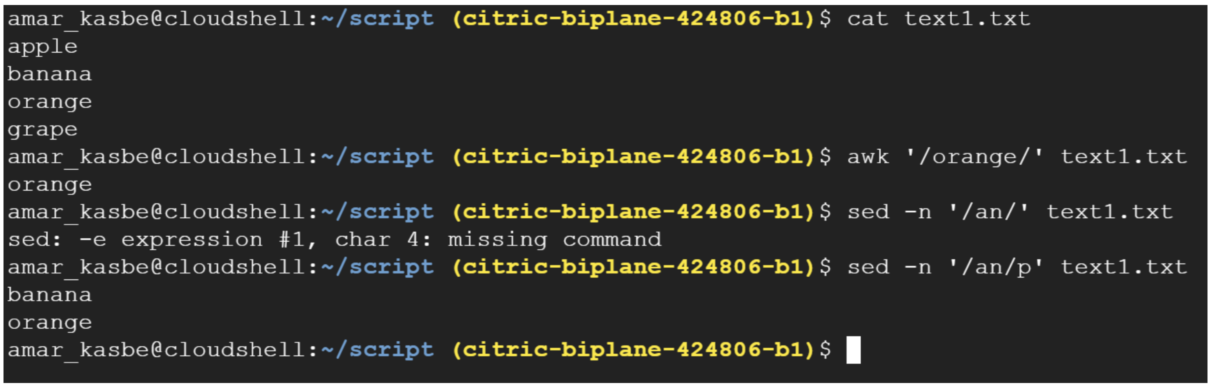
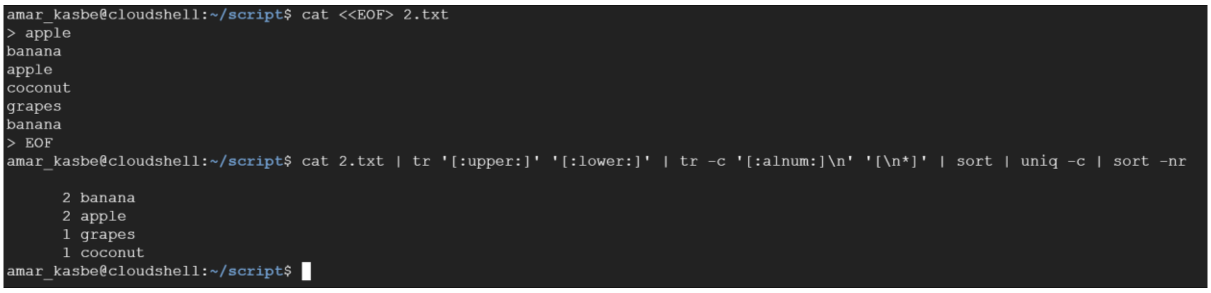
Shell Script Assignment 5

1. How Can We Find a Pattern in a File Without Using the *grep* Command?



```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ cat text1.txt
apple
banana
orange
grape
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ awk '/orange/' text1.txt
orange
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed -n '/an/' text1.txt
sed: -e expression #1, char 4: missing command
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed -n '/an/p' text1.txt
banana
orange
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

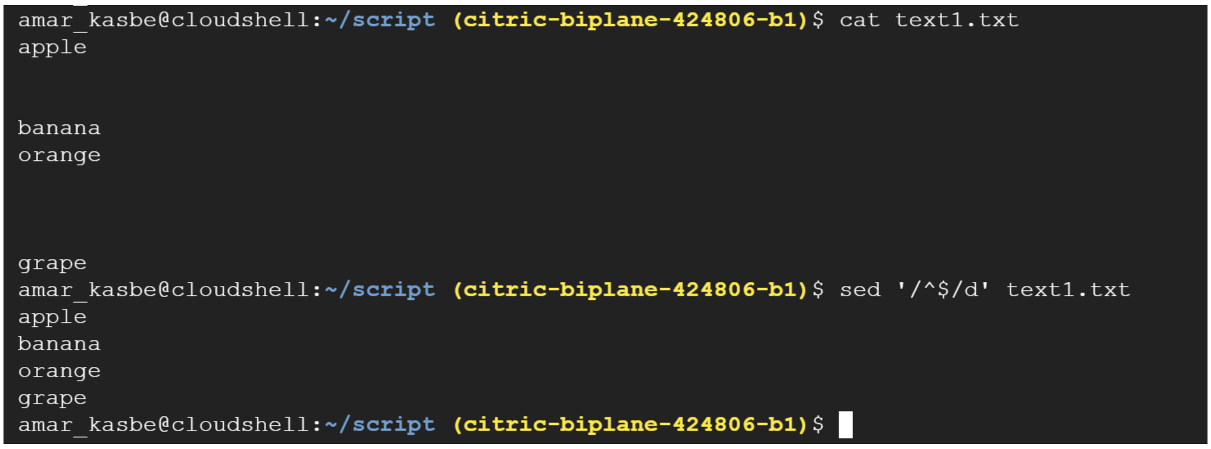
2. Given a file, write a command sequence to find the count of each word.



```
amar_kasbe@cloudshell:~/script$ cat <<EOF> 2.txt
> apple
banana
apple
coconut
grapes
banana
> EOF
amar_kasbe@cloudshell:~/script$ cat 2.txt | tr '[:upper:]' '[:lower:]' | tr -c '[:alnum:]\n' '[\n*]' | sort | uniq -c | sort -nr

      2 banana
      2 apple
      1 grapes
      1 coconut
amar_kasbe@cloudshell:~/script$
```

3. How do we delete all blank lines in a file?

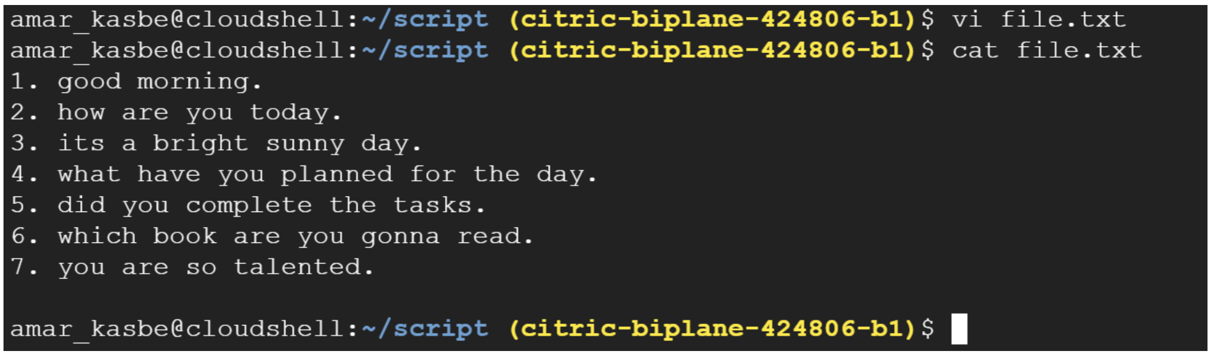
➔ /^$/ is a regular expression that matches blank lines.



```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ cat text1.txt
apple


banana
orange




grape
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed '/^$/d' text1.txt
apple
banana
orange
grape
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

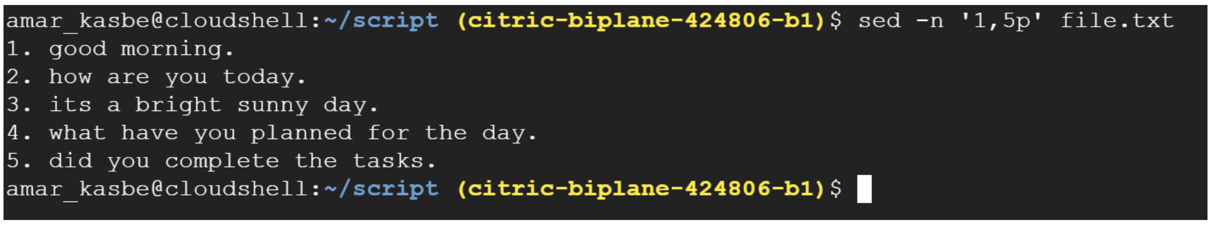
4. Create a file1.txt that consists of following

1. hello good morning.
2. how are you today.
3. its a bright sunny day.
4. what have you planned for the day.
5. did you complete the tasks.
6. which book are you gonna read. you are so talented.



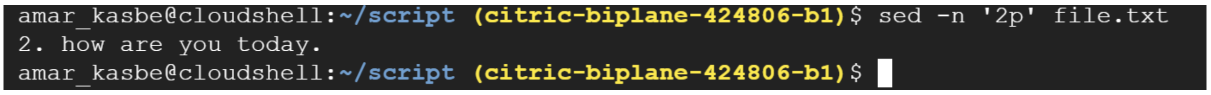
```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ vi file.txt
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ cat file.txt
1. good morning.
2. how are you today.
3. its a bright sunny day.
4. what have you planned for the day.
5. did you complete the tasks.
6. which book are you gonna read.
7. you are so talented.

amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

➔ Display first 5 lines using sed



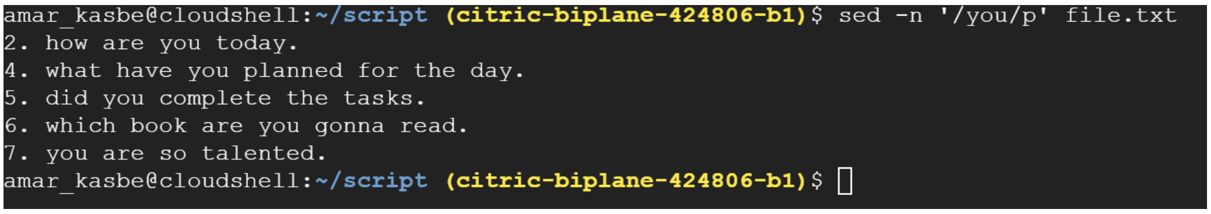
```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed -n '1,5p' file.txt
1. good morning.
2. how are you today.
3. its a bright sunny day.
4. what have you planned for the day.
5. did you complete the tasks.
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

➔ Display only 2nd line using sed



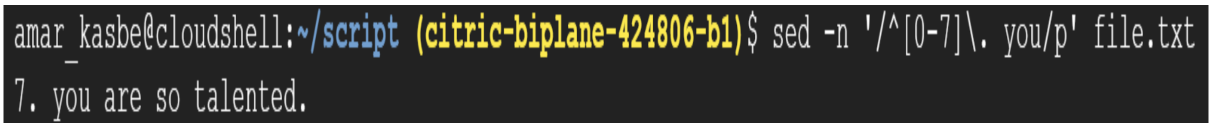
```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed -n '2p' file.txt
2. how are you today.
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

➔ Display all the lines that consists of pattern you



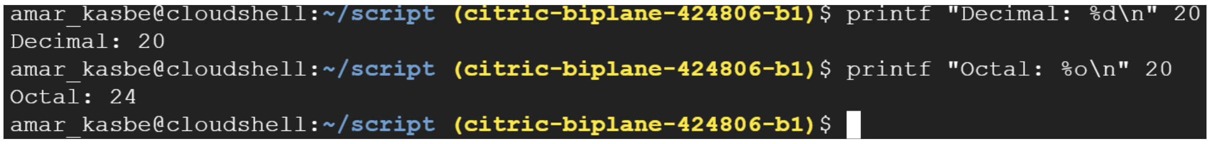
```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed -n '/you/p' file.txt
2. how are you today.
4. what have you planned for the day.
5. did you complete the tasks.
6. which book are you gonna read.
7. you are so talented.
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

➔ Display all the lines that begin with the pattern you



```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sed -n '/^[0-7]\. you/p' file.txt
7. you are so talented.
```

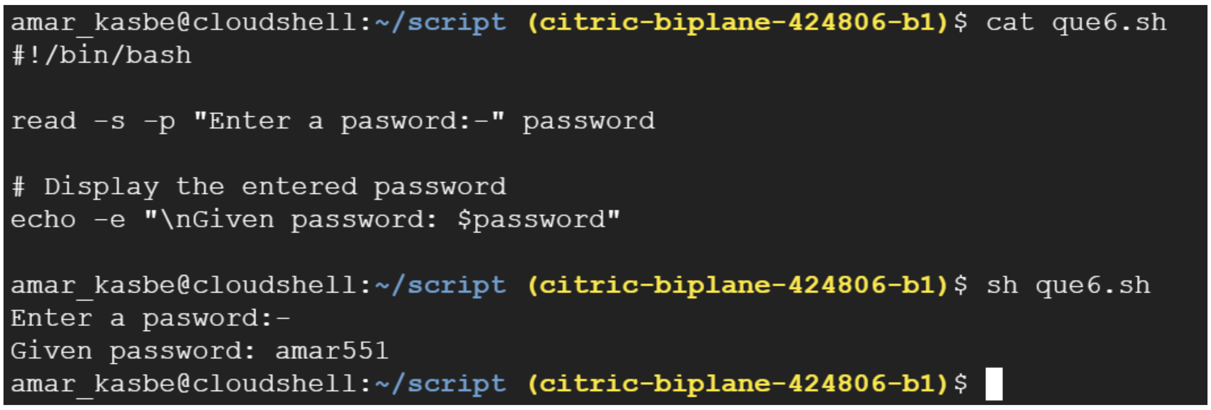
5. Print the decimal and octal value of 20 using printf



```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ printf "Decimal: %d\n" 20
Decimal: 20
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ printf "Octal: %o\n" 20
Octal: 24
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

6. Write a shell script for reading the password from user but display is disabled

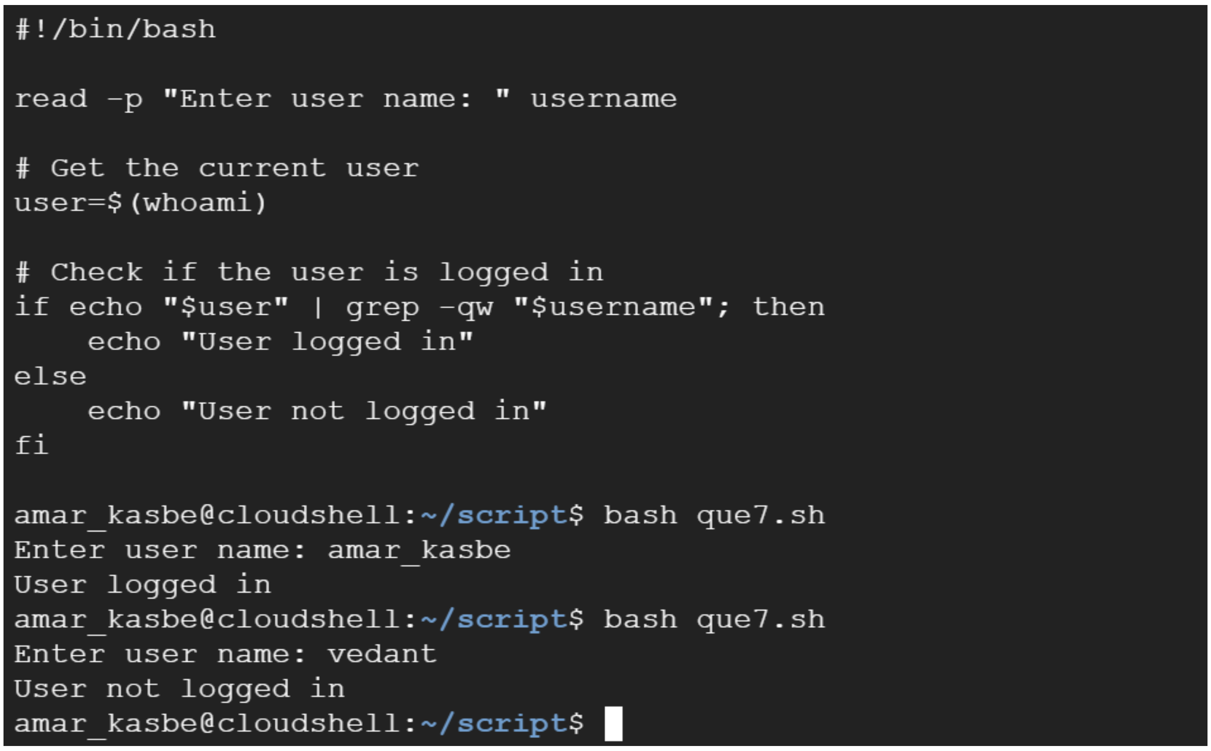
Expected Output:
enter a password
given password=cloudera



```
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ cat que6.sh
#!/bin/bash

read -s -p "Enter a pasword:-" password

# Display the entered password
echo -e "\nGiven password: $password"

amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$ sh que6.sh
Enter a pasword:-
Given password: amar551
amar_kasbe@cloudshell:~/script (citric-biplane-424806-b1)$
```

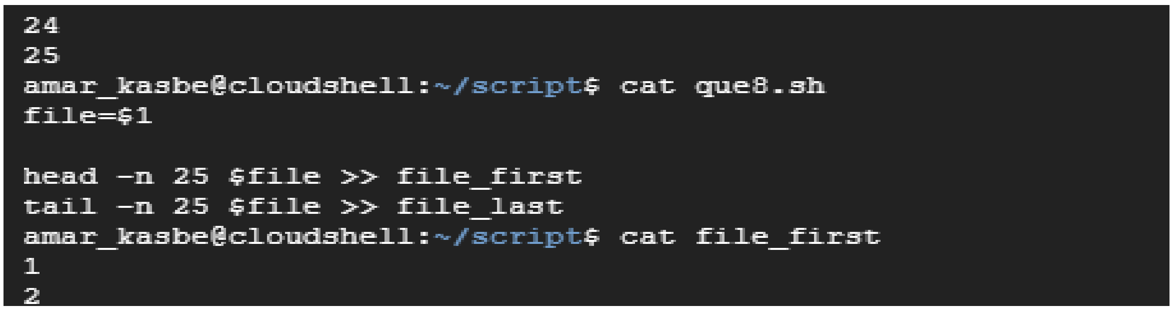
7. write a shell script using who command and grep that asks the user to enter username and displays the output as User Logged In or NOT respectively.

Expected Output1:
enter user name: cloudera
user not logged in
Expected Output2:
enter user name: farha
user logged in

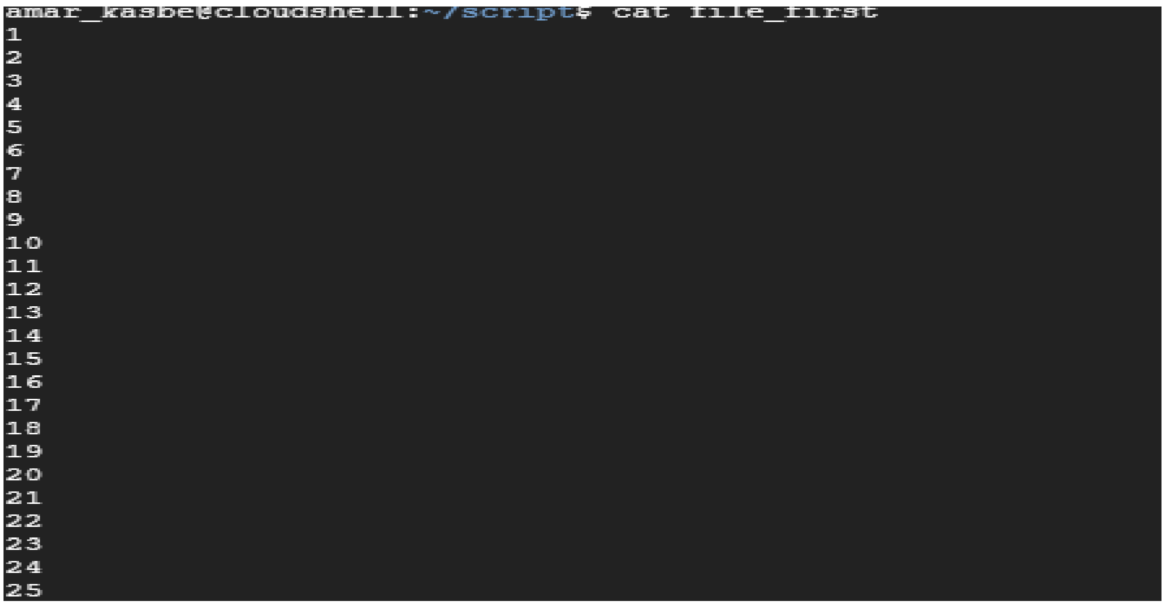


```
#!/bin/bash

read -p "Enter user name: " username

# Get the current user
user=$(whoami)

# Check if the user is logged in
if echo "$user" | grep -qw "$username"; then
    echo "User logged in"
else
    echo "User not logged in"
fi

amar_kasbe@cloudshell:~/script$ bash que7.sh
Enter user name: amar_kasbe
User logged in
amar_kasbe@cloudshell:~/script$ bash que7.sh
Enter user name: vedant
User not logged in
amar_kasbe@cloudshell:~/script$
```

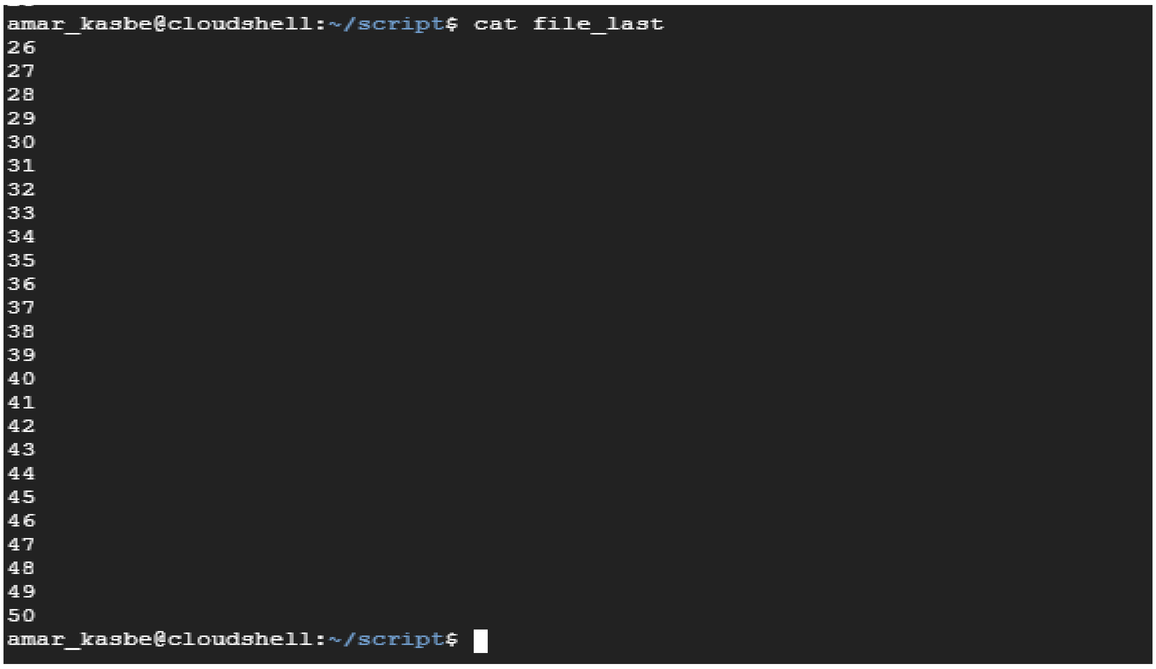
8. A file consists of 50 lines. Write a shell script to split into two files having 25 lines each.



```
24
25
amar_kasbe@cloudshell:~/script$ cat que8.sh
file=$1

head -n 25 $file >> file_first
tail -n 25 $file >> file_last
amar_kasbe@cloudshell:~/script$ cat file_first
1
2
```

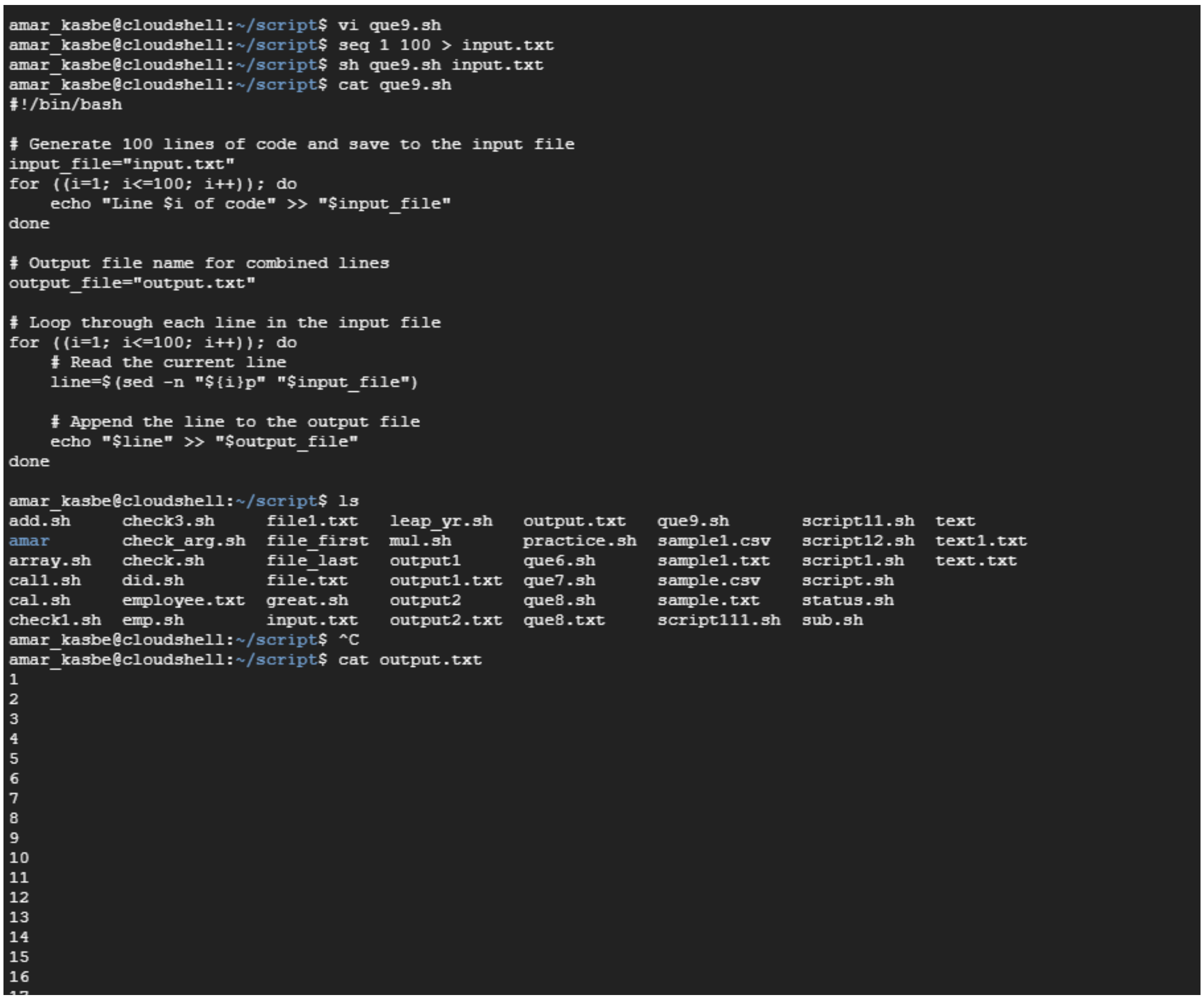


```
amar_kasbe@cloudshell:~/script$ cat file_first
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```



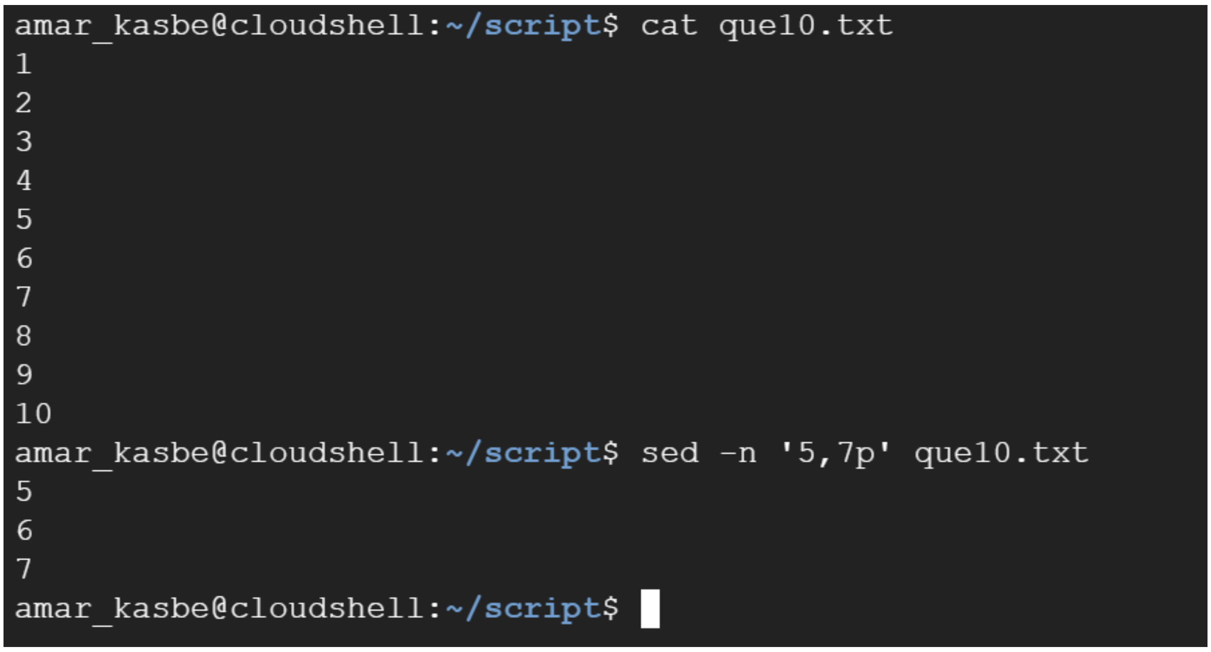
```
amar_kasbe@cloudshell:~/script$ cat file_last
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
amar_kasbe@cloudshell:~/script$
```

9. A file has 100 lines. Use for loop to read each line and append into another file.



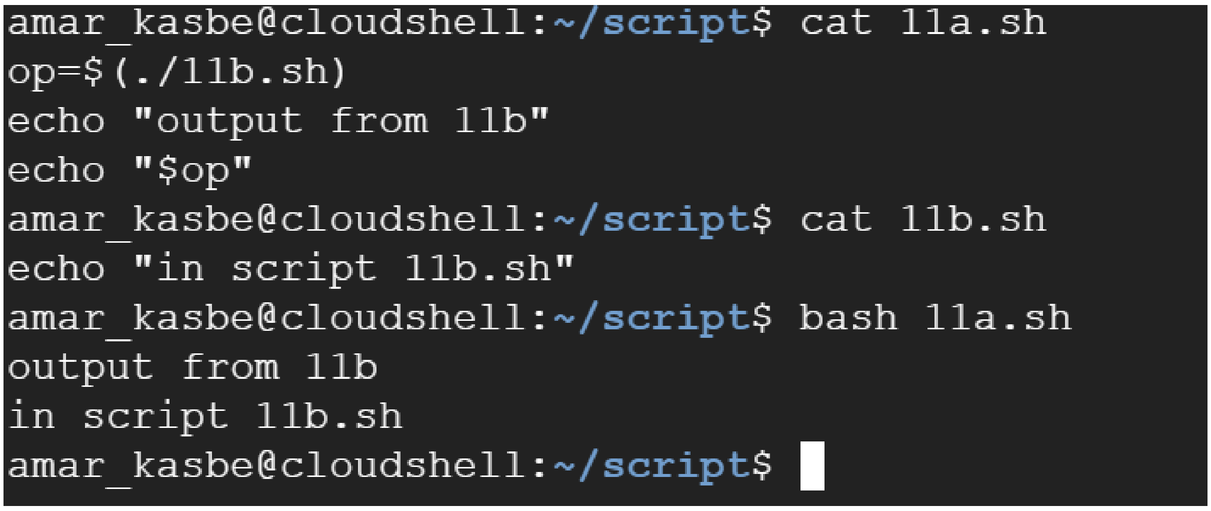
```
amar_kasbe@cloudshell:~/script$ vi que9.sh
amar_kasbe@cloudshell:~/script$ seq 1 100 > input.txt
amar_kasbe@cloudshell:~/script$ sh que9.sh input.txt
amar_kasbe@cloudshell:~/script$ cat que9.sh
#!/bin/bash

# Generate 100 lines of code and save to the input file
input_file="input.txt"
for ((i=1; i<=100; i++)); do
    echo "Line $i of code" >> "$input_file"
done

# Output file name for combined lines
output_file="output.txt"

# Loop through each line in the input file
for ((i=1; i<=100; i++)); do
    # Read the current line
    line=$(sed -n "${i}p" "$input_file")

    # Append the line to the output file
    echo "$line" >> "$output_file"
done

amar_kasbe@cloudshell:~/script$ ls
add.sh     check3.sh     file1.txt   leap_yr.sh   output.txt   que9.sh      script11.sh  text
amar       check_arg.sh  file_first  mul.sh       practice.sh  sample1.csv  script12.sh  text1.txt
array.sh   check.sh      file_last   output1      que6.sh      sample1.txt  script1.sh   text.txt
call.sh    did.sh        file.txt    output1.txt  que7.sh      sample.csv   script.sh
cal.sh     employee.txt  great.sh    output2      que8.sh      sample.txt   status.sh
check1.sh  emp.sh        input.txt   output2.txt  que8.txt     script111.sh sub.sh
amar_kasbe@cloudshell:~/script$ ^C
amar_kasbe@cloudshell:~/script$ cat output.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
```

10. A file has 10 lines. Print 5th to 7th line.



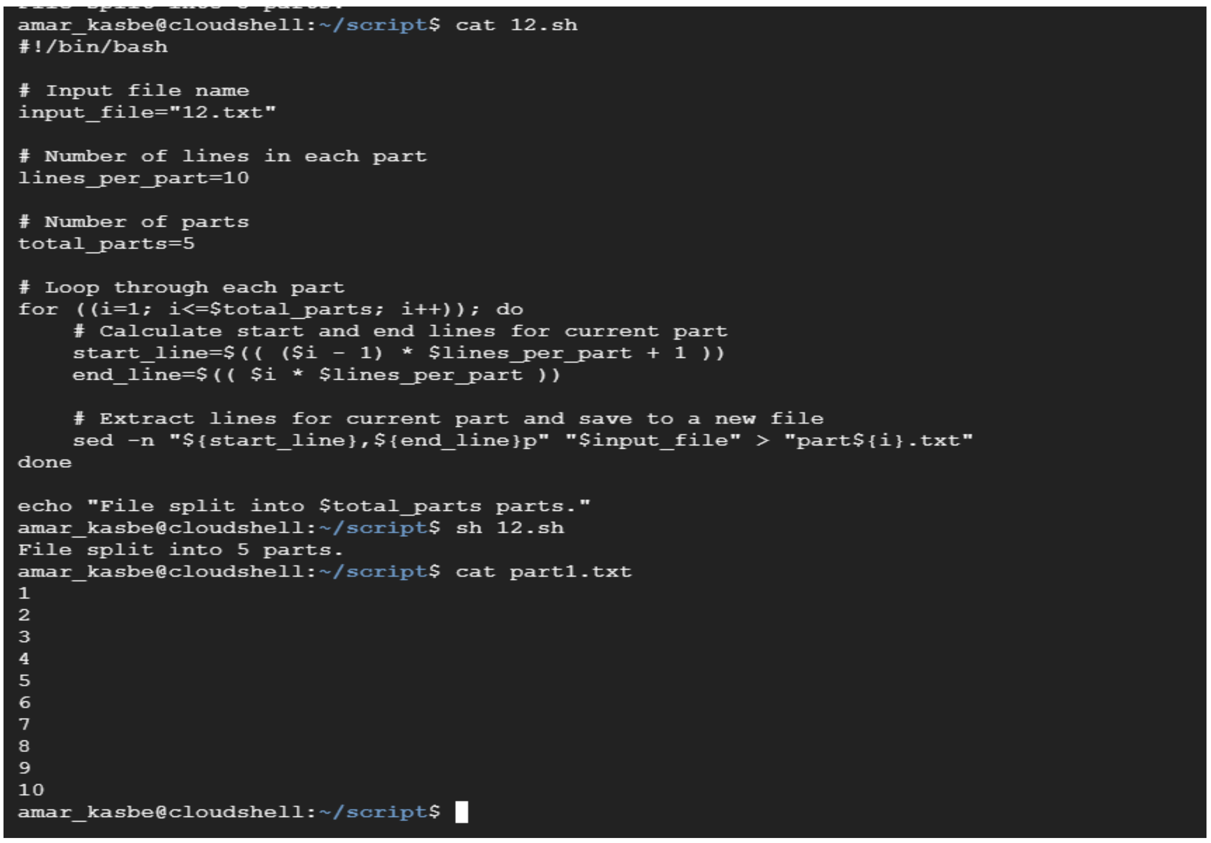
```
amar_kasbe@cloudshell:~/script$ cat que10.txt
1
2
3
4
5
6
7
8
9
10
amar_kasbe@cloudshell:~/script$ sed -n '5,7p' que10.txt
5
6
7
amar_kasbe@cloudshell:~/script$
```

11. Calling one shell script from another shell script. Also capture the output

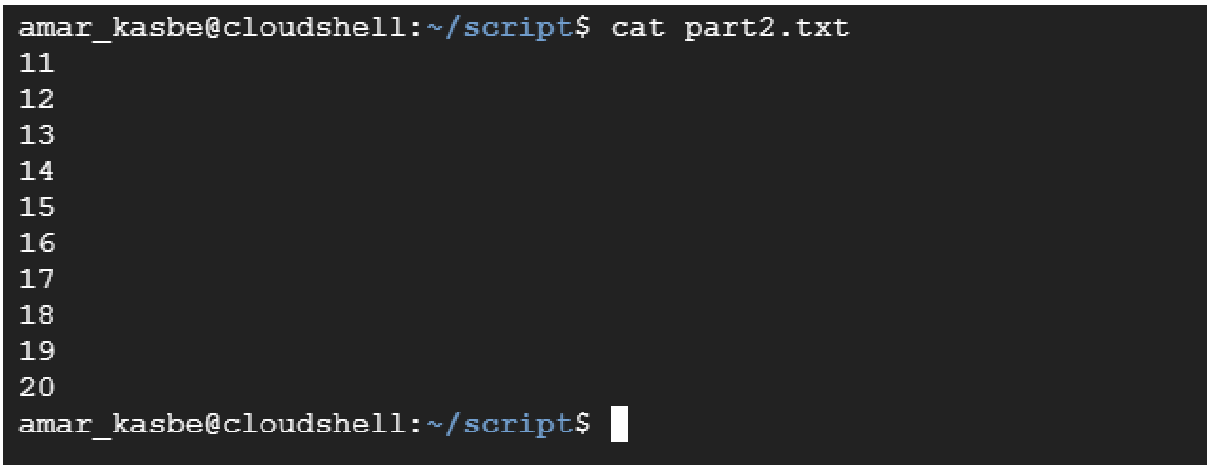


```
amar_kasbe@cloudshell:~/script$ cat 11a.sh
op=$(./11b.sh)
echo "output from 11b"
echo "$op"
amar_kasbe@cloudshell:~/script$ cat 11b.sh
echo "in script 11b.sh"
amar_kasbe@cloudshell:~/script$ bash 11a.sh
output from 11b
in script 11b.sh
amar_kasbe@cloudshell:~/script$
```

12. Splitting a file into multiple files using for loop



```
amar_kasbe@cloudshell:~/script$ cat 12.sh
#!/bin/bash

# Input file name
input_file="12.txt"

# Number of lines in each part
lines_per_part=10

# Number of parts
total_parts=5

# Loop through each part
for ((i=1; i<=$total_parts; i++)); do
    # Calculate start and end lines for current part
    start_line=$(( ($i - 1) * $lines_per_part + 1 ))
    end_line=$(( $i * $lines_per_part ))

    # Extract lines for current part and save to a new file
    sed -n "${start_line},${end_line}p" "$input_file" > "part${i}.txt"
done

echo "File split into $total_parts parts."
amar_kasbe@cloudshell:~/script$ sh 12.sh
File split into 5 parts.
amar_kasbe@cloudshell:~/script$ cat part1.txt
1
2
3
4
5
6
7
8
9
10
amar_kasbe@cloudshell:~/script$
```



```
amar_kasbe@cloudshell:~/script$ cat part2.txt
11
12
13
14
15
16
17
18
19
20
amar_kasbe@cloudshell:~/script$
```