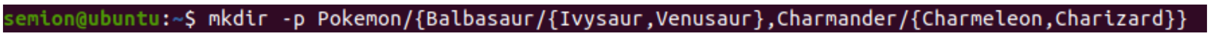
Linux:

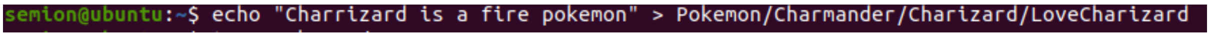
● Create directories as shown in the drawing by minimum commands.



```
semion@ubuntu:~$ mkdir -p Pokemon/{Balbasaur/{Ivysaur,Venusaur},Charmander/{Charmeleon,Charizard}}
```

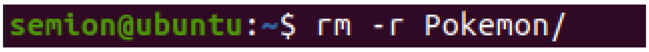
● Echo “Charizard is a fire pokemon” in “LoveCharizard” file under “Charizard”

subdir.



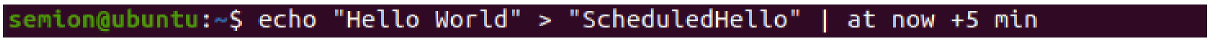
```
semion@ubuntu:~$ echo "Charrizard is a fire pokemon" > Pokemon/Charmander/Charizard/LoveCharizard
```

● Delete Pokemon file.



```
semion@ubuntu:~$ rm -r Pokemon/
```

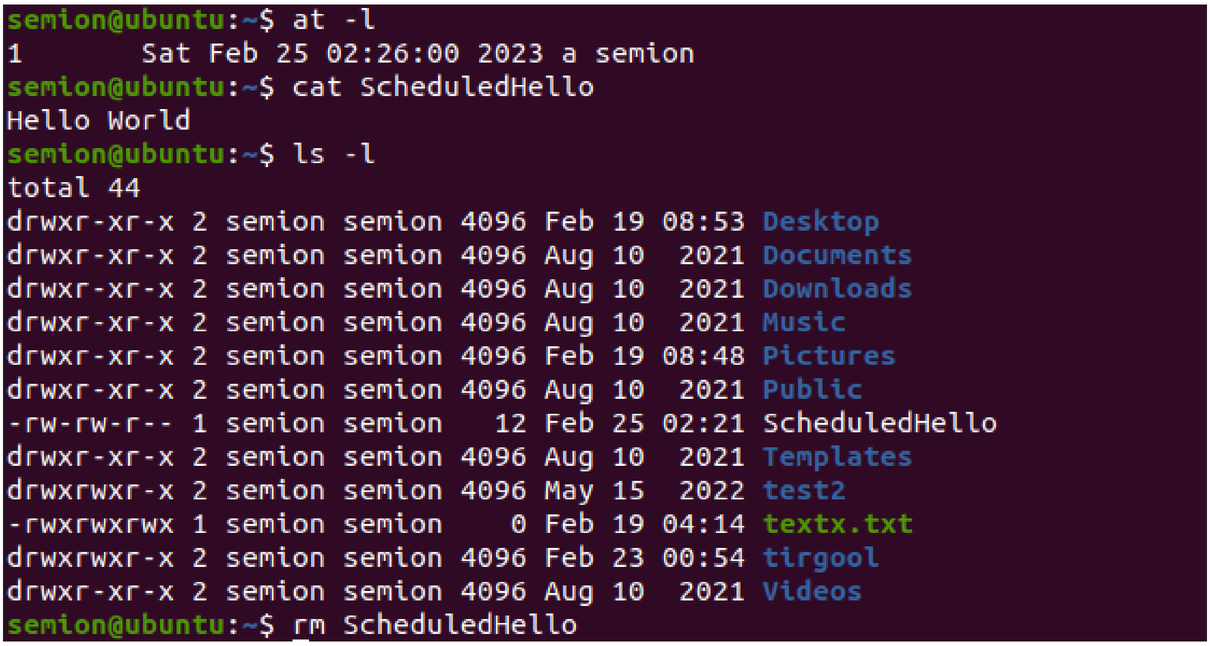
● Echo “Hello World” in “ScheduledHello” after 5 minutes from now.



```
semion@ubuntu:~$ echo "Hello World" > "ScheduledHello" | at now +5 min
```

● check after 5 minutes if the file “ScheduledHello” exists in your file.

● Delete “ScheduledHello” file.

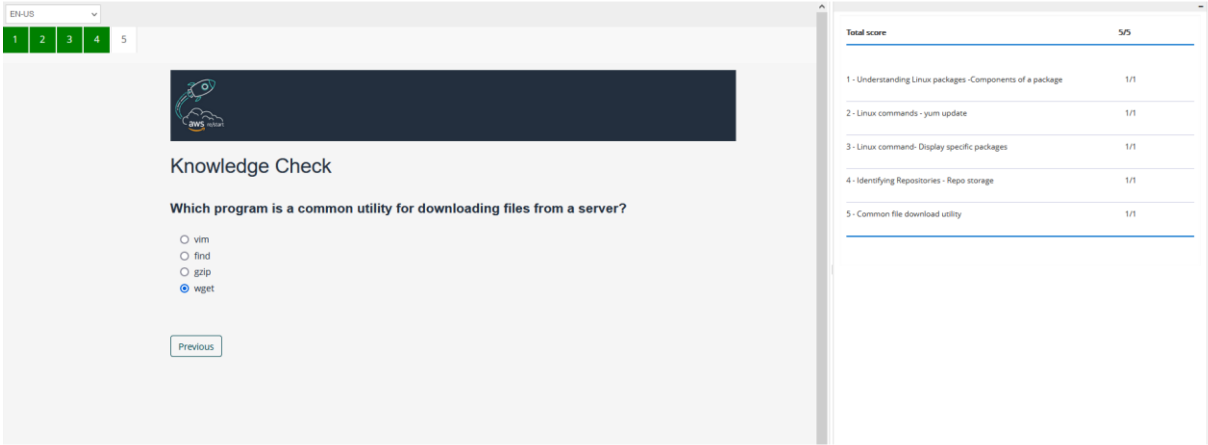


```
semion@ubuntu:~$ at -l
1       Sat Feb 25 02:26:00 2023 a semion
semion@ubuntu:~$ cat ScheduledHello
Hello World
semion@ubuntu:~$ ls -l
total 44
drwxr-xr-x 2 semion semion 4096 Feb 19 08:53 Desktop
drwxr-xr-x 2 semion semion 4096 Aug 10  2021 Documents
drwxr-xr-x 2 semion semion 4096 Aug 10  2021 Downloads
drwxr-xr-x 2 semion semion 4096 Aug 10  2021 Music
drwxr-xr-x 2 semion semion 4096 Feb 19 08:48 Pictures
drwxr-xr-x 2 semion semion 4096 Aug 10  2021 Public
-rw-rw-r-- 1 semion semion   12 Feb 25 02:21 ScheduledHello
drwxr-xr-x 2 semion semion 4096 Aug 10  2021 Templates
drwxrwxr-x 2 semion semion 4096 May 15  2022 test2
-rwxrwxrwx 1 semion semion    0 Feb 19 04:14 textx.txt
drwxrwxr-x 2 semion semion 4096 Feb 23 00:54 tirgool
drwxr-xr-x 2 semion semion 4096 Aug 10  2021 Videos
semion@ubuntu:~$ rm ScheduledHello
```

===

Canvas:

Pay attention: search every lab by its title (the numbers aren't relevant)

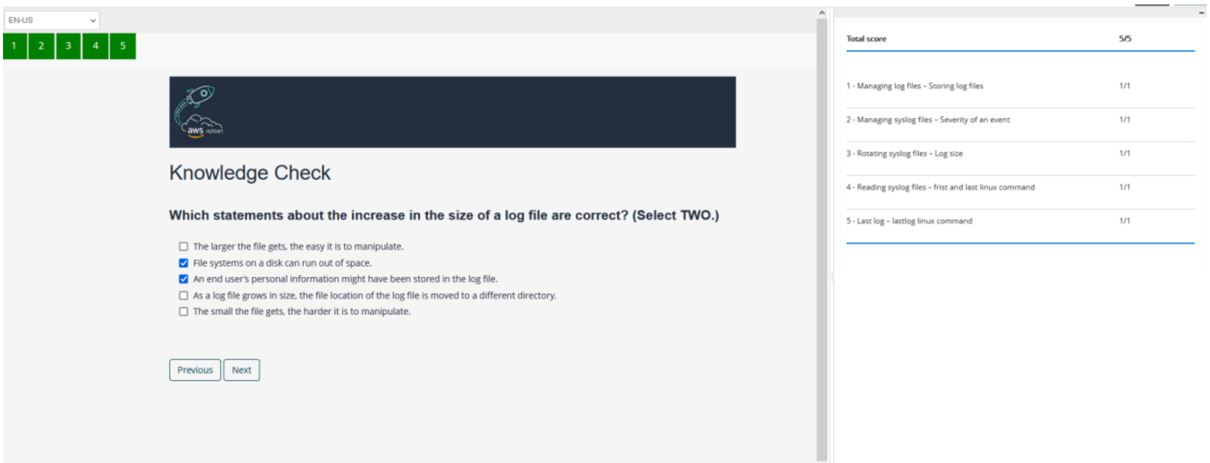


Knowledge Check

Which program is a common utility for downloading files from a server?

○ vim
○ find
○ gzip
● wget

Total score	5/5
1 - Understanding Linux packages -Components of a package	1/1
2 - Linux commands - yum update	1/1
3 - Linux command- Display specific packages	1/1
4 - Identifying Repositories - Repo storage	1/1
5 - Common file download utility	1/1

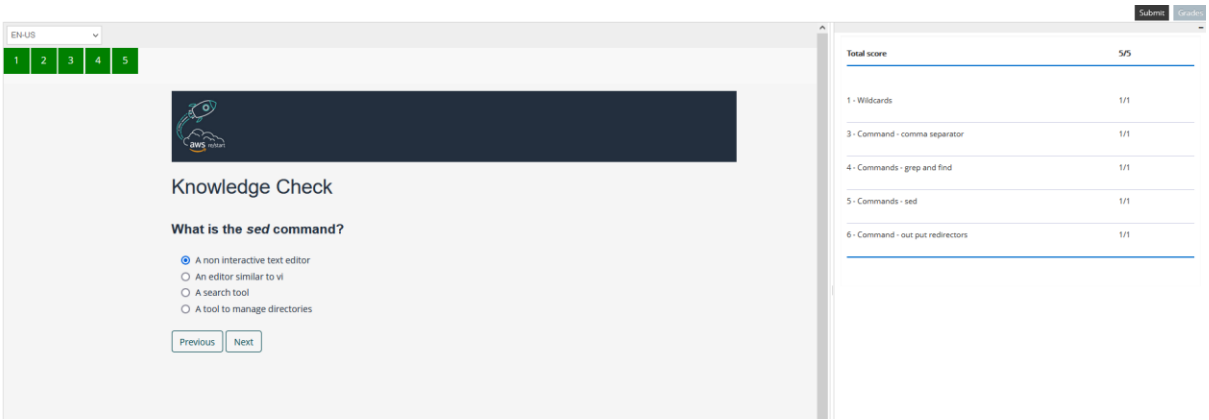


Knowledge Check

Which statements about the increase in the size of a log file are correct? (Select TWO.)

☐ The larger the file gets, the easy it is to manipulate.
☑ File systems on a disk can run out of space.
☑ An end user's personal information might have been stored in the log file.
☐ As a log file grows in size, the file location of the log file is moved to a different directory.
☐ The small the file gets, the harder it is to manipulate.

Total score	5/5
1 - Managing log files – Storing log files	1/1
2 - Managing syslog files – Severity of an event	1/1
3 - Rotating syslog files – Log size	1/1
4 - Reading syslog files – frist and last linux command	1/1
5 - Last log – lastlog linux command	1/1

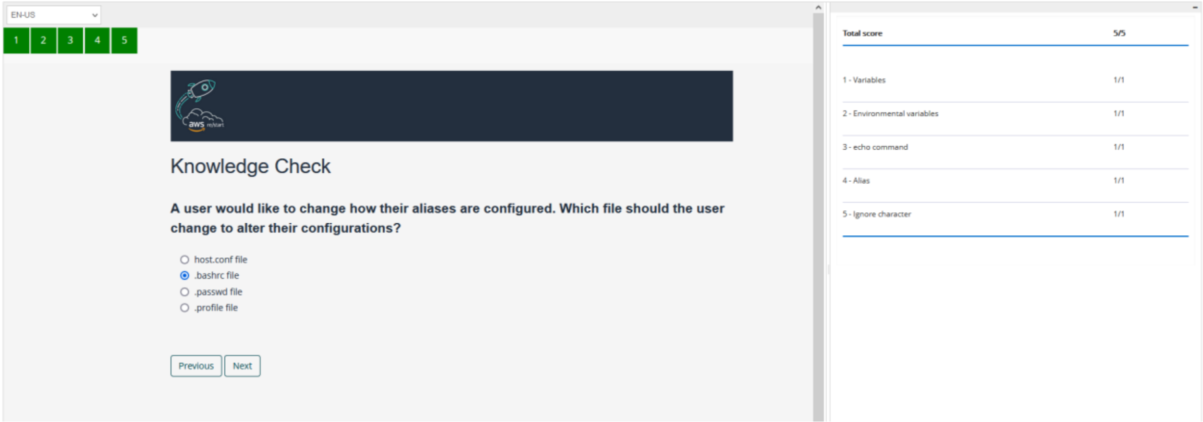


Knowledge Check

What is the *sed* command?

● A non interactive text editor
○ An editor similar to vi
○ A search tool
○ A tool to manage directories

Total score	5/5
1 - Wildcards	1/1
3 - Command - comma separator	1/1
4 - Commands - grep and find	1/1
5 - Commands - sed	1/1
6 - Command - out put redirectors	1/1



Knowledge Check

A user would like to change how their aliases are configured. Which file should the user change to alter their configurations?

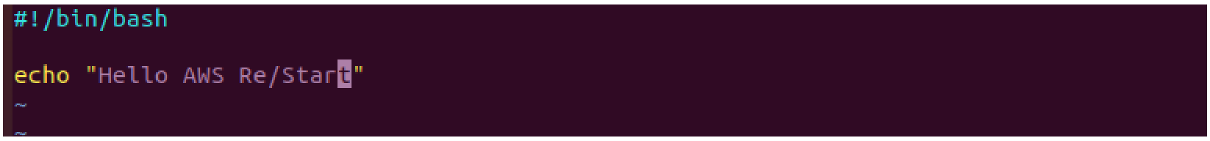
○ host.conf file
● .bashrc file
○ .passwd file
○ .profile file

Total score	5/5
1 - Variables	1/1
2 - Environmental variables	1/1
3 - echo command	1/1
4 - Alias	1/1
5 - Ignore character	1/1

===

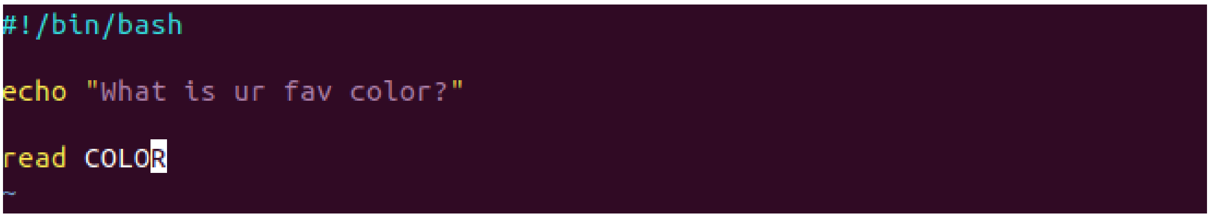
Bash:

1. Write a script that prints the message " hello AWS Re/Start "



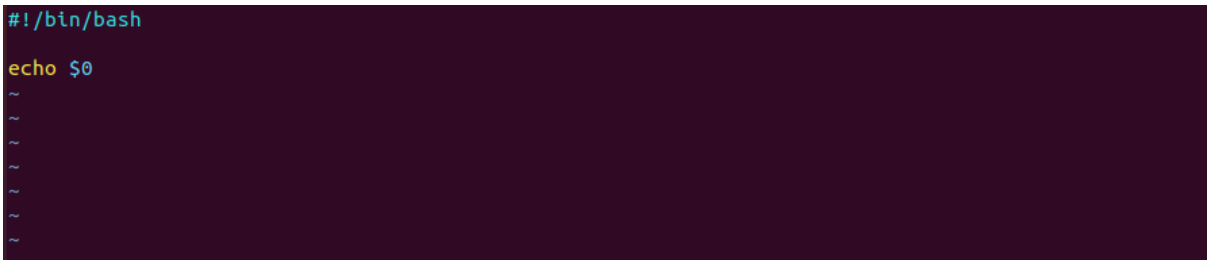
```
#!/bin/bash

echo "Hello AWS Re/Start"
~
~
```

2. Write a script that ask the user his favorite color and print it.



```
#!/bin/bash

echo "What is ur fav color?"

read COLOR
~
```

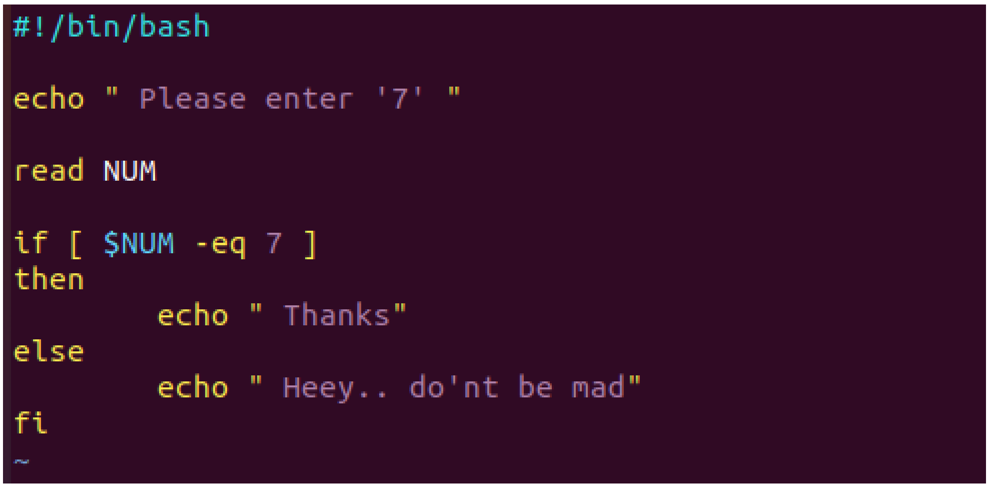
3. Execute script that echos your script file name.



```
#!/bin/bash

echo $0
~
~
~
~
~
~
~
```

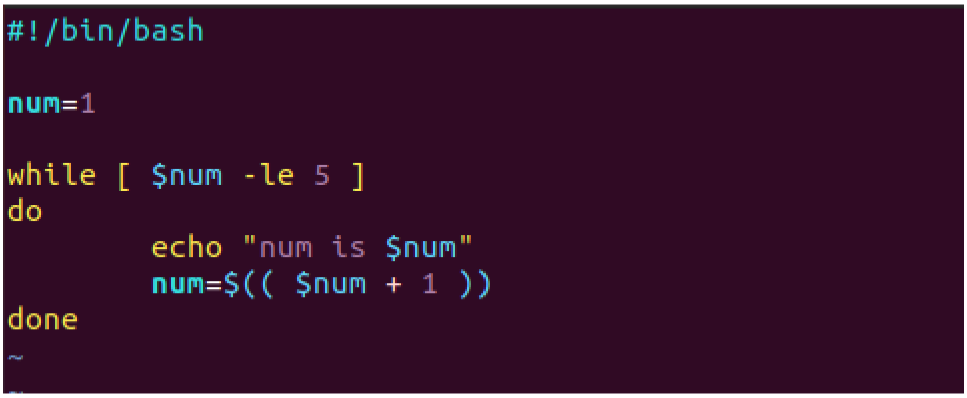
4. Script - ask the user to enter the number “7” , if yes echos “Thanks” , if no echos

“Heey .. don’t be mad”



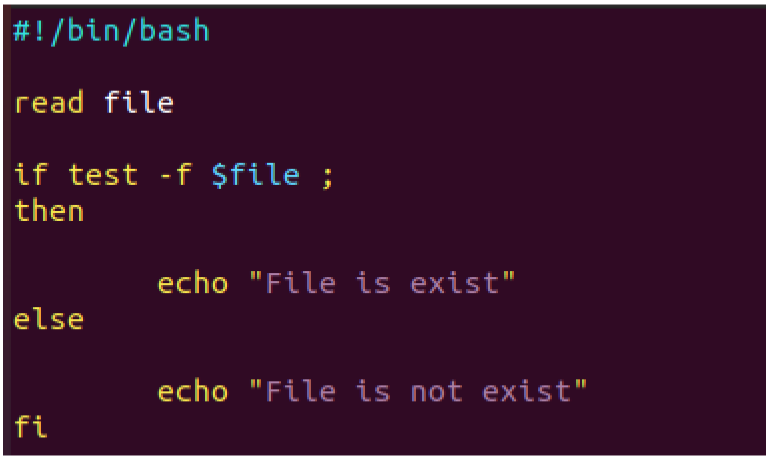
```
#!/bin/bash

echo " Please enter '7' "

read NUM

if [ $NUM -eq 7 ]
then
        echo " Thanks"
else
        echo " Heey.. do'nt be mad"
fi
~
```

5. Write a script that display numbers from 1 to 5 line by line (hint: While)



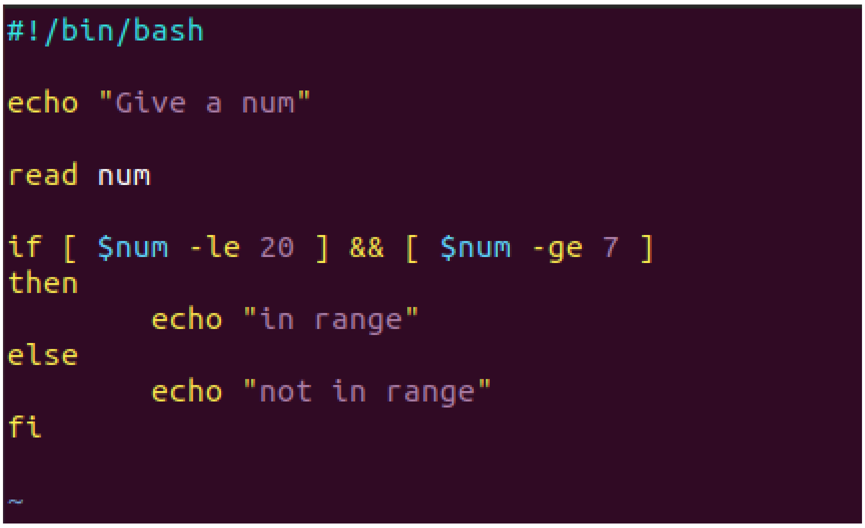
```
#!/bin/bash

num=1

while [ $num -le 5 ]
do
        echo "num is $num"
        num=$(( $num + 1 ))
done
~
```

6. Write a script that prompts the user to enter a filename, and then checks whether the file exists.



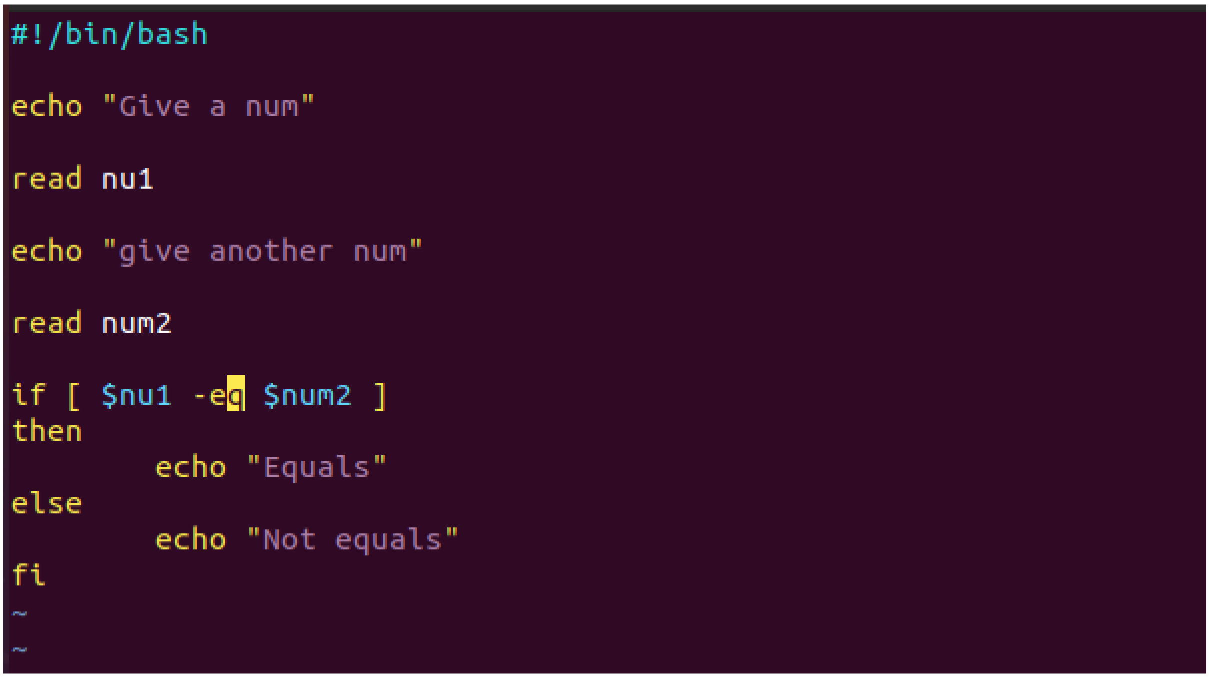
```
#!/bin/bash

read file

if test -f $file ;
then

        echo "File is exist"
else

        echo "File is not exist"
fi
```

7. Write a script that prompts the user to enter a number between 7 and 20, and then checks whether the number is within the specified range.



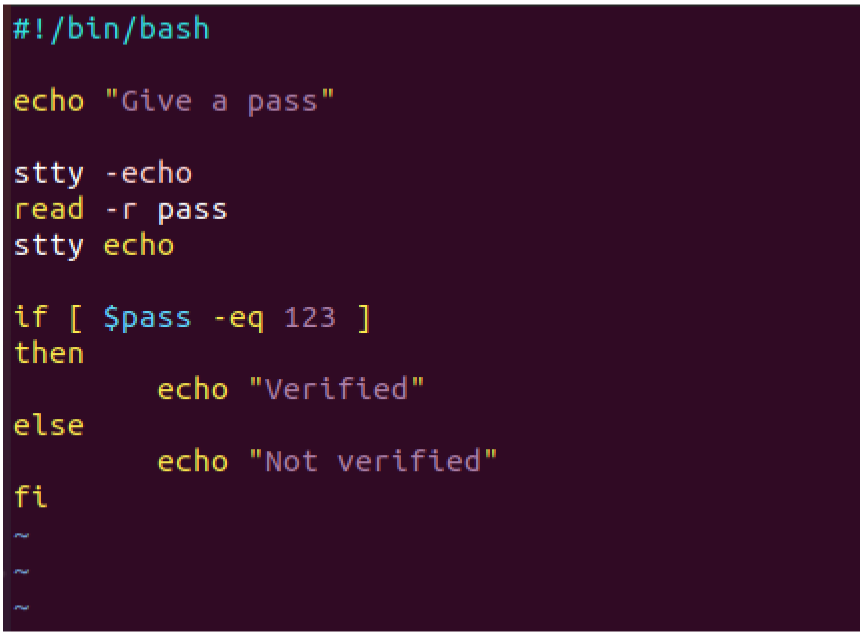
```
#!/bin/bash

echo "Give a num"

read num

if [ $num -le 20 ] && [ $num -ge 7 ]
then
        echo "in range"
else
        echo "not in range"
fi

~
```

8. Write a script that request the user to enter two numbers and compare between them, if they are equal echos “equal” otherwise echos “not equal.



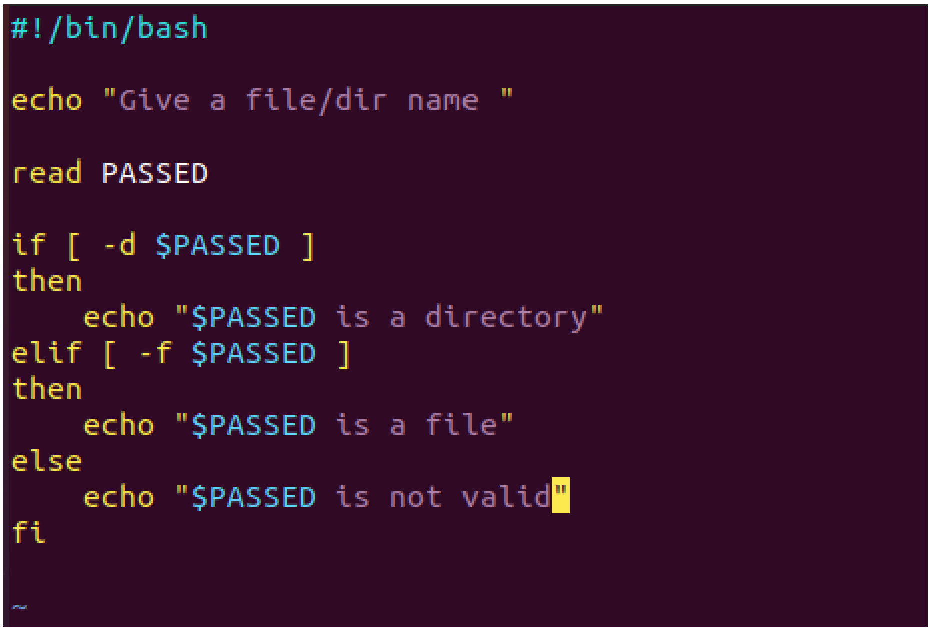
```
#!/bin/bash

echo "Give a num"

read nu1

echo "give another num"

read num2

if [ $nu1 -eq $num2 ]
then
        echo "Equals"
else
        echo "Not equals"
fi
~
~
```

9. Ask the user to enter a password. Check if the password matches “123”. If it matches echo verified. If not, echo denied.(use silent mode with read)



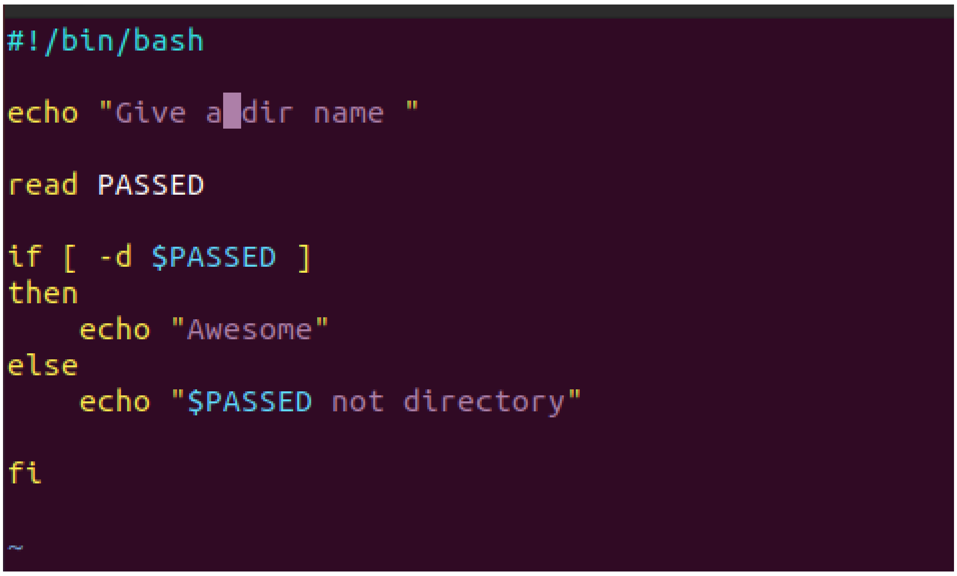
```
#!/bin/bash

echo "Give a pass"

stty -echo
read -r pass
stty echo

if [ $pass -eq 123 ]
then
        echo "Verified"
else
        echo "Not verified"
fi
~
~
~
```

10. write a script that request the name of a file or directory (in your current directory) and check if it is a regular file/dir . Echos “Great” if it’s regular file otherwise “not regular file”



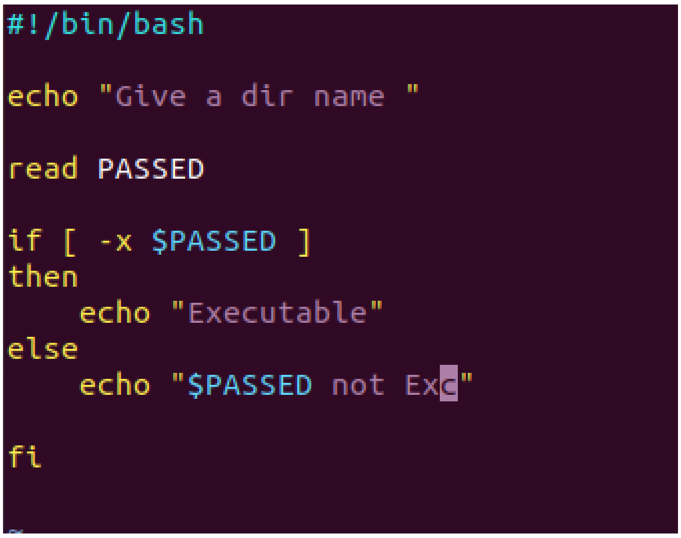
```
#!/bin/bash

echo "Give a file/dir name "

read PASSED

if [ -d $PASSED ]
then
    echo "$PASSED is a directory"
elif [ -f $PASSED ]
then
    echo "$PASSED is a file"
else
    echo "$PASSED is not valid"
fi

~
```

11. write a script that request the name of a file or directory and check if it is a directory, Echos “Awesome” if it’s directory. Otherwise”not directory”.



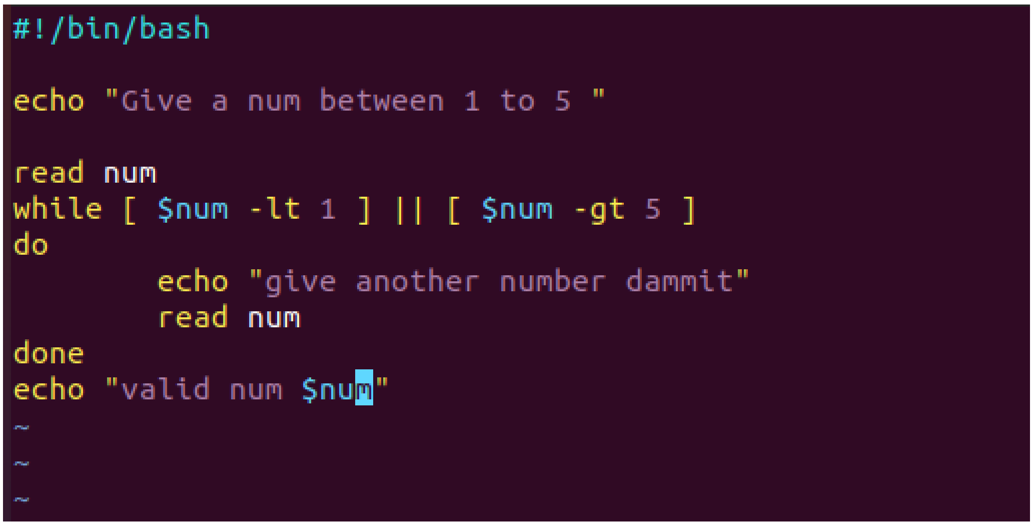
```
#!/bin/bash

echo "Give a dir name "

read PASSED

if [ -d $PASSED ]
then
    echo "Awesome"
else
    echo "$PASSED not directory"

fi

~
```

12. write a script that request the name of a file or directory and check if it is a executable, Echos “it’s executable” if it’s executable otherwise “not executable”.



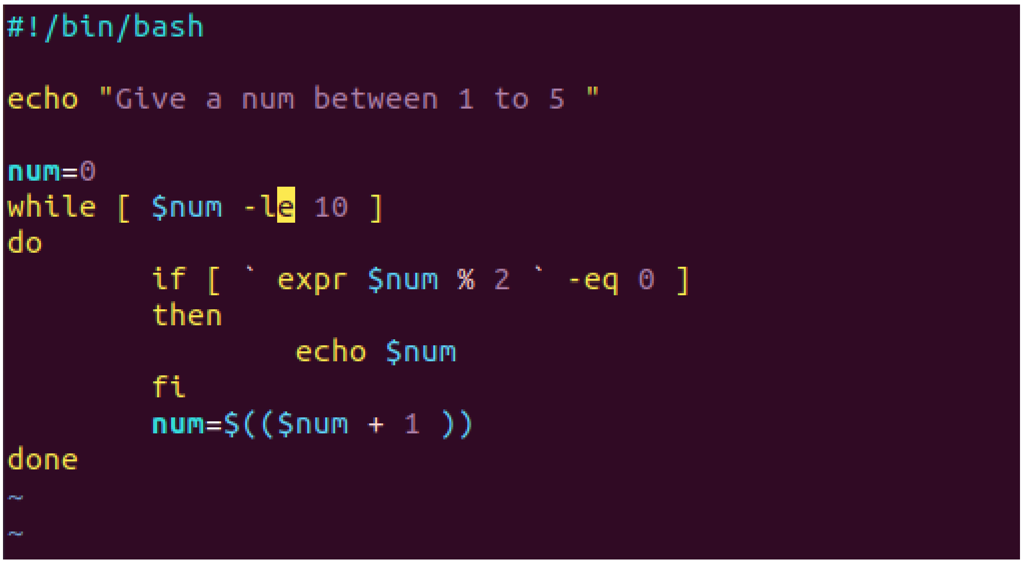
```
#!/bin/bash

echo "Give a dir name "

read PASSED

if [ -x $PASSED ]
then
    echo "Executable"
else
    echo "$PASSED not Exc"

fi
```

13. Write a while loop that prompts the user to enter a number between 1 and 5. The loop should continue until the user enters a valid number.



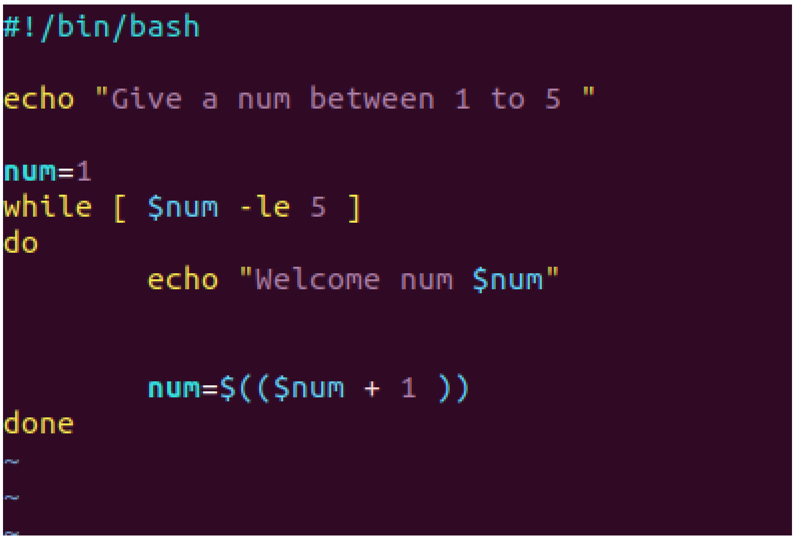
```
#!/bin/bash

echo "Give a num between 1 to 5 "

read num
while [ $num -lt 1 ] || [ $num -gt 5 ]
do
        echo "give another number dammit"
        read num
done
echo "valid num $num"
~
~
~
```

14. Write a while loop that prints the first 10 even numbers.



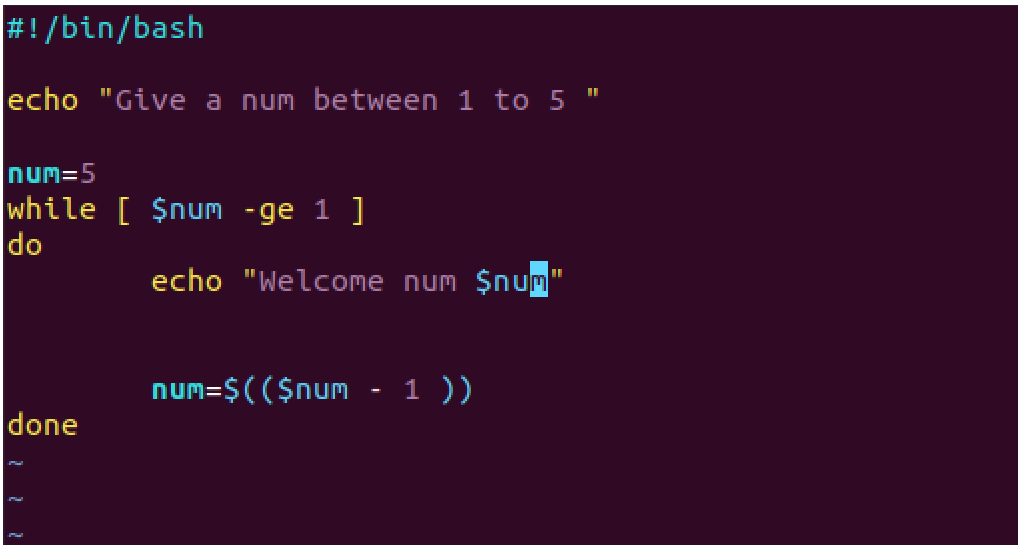
```
#!/bin/bash

echo "Give a num between 1 to 5 "

num=0
while [ $num -le 10 ]
do
        if [ ` expr $num % 2 ` -eq 0 ]
        then
                echo $num
        fi
        num=$(($num + 1 ))
done
~
~
```

15. write a while loop that echos welcome and the iteration number (for example, welcome 1, welcome 2...) runs 5 times, the print should start at 1 and finish at 5.



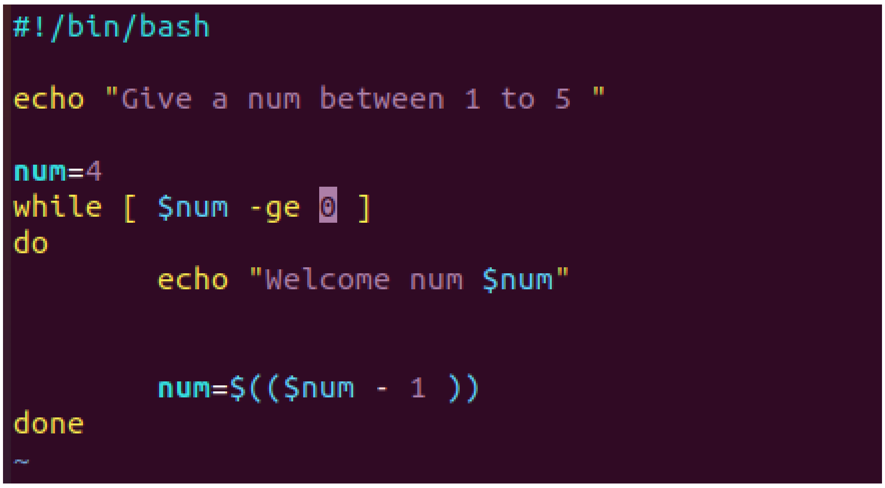
```
#!/bin/bash

echo "Give a num between 1 to 5 "

num=1
while [ $num -le 5 ]
do
        echo "Welcome num $num"


        num=$(($num + 1 ))
done
~
~
~
```

16. write a while loop that echos welcome and the iteration number (for example, welcome 1, welcome 2...) runs 5 times, the print should start at 5 and finish at 1.



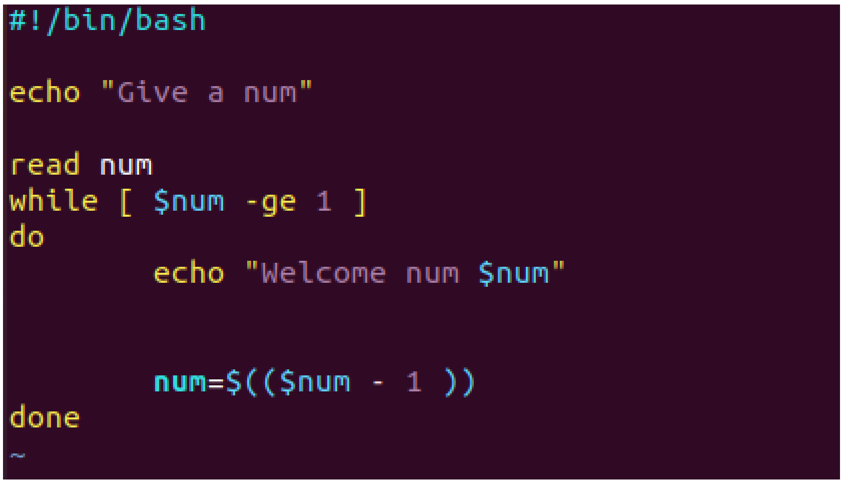
```
#!/bin/bash

echo "Give a num between 1 to 5 "

num=5
while [ $num -ge 1 ]
do
        echo "Welcome num $num"


        num=$(($num - 1 ))
done
~
~
~
```

17. create a while loop which echoes welcome and the iteration number and runs 5 times . the printout should start at 4 and finish at 0.



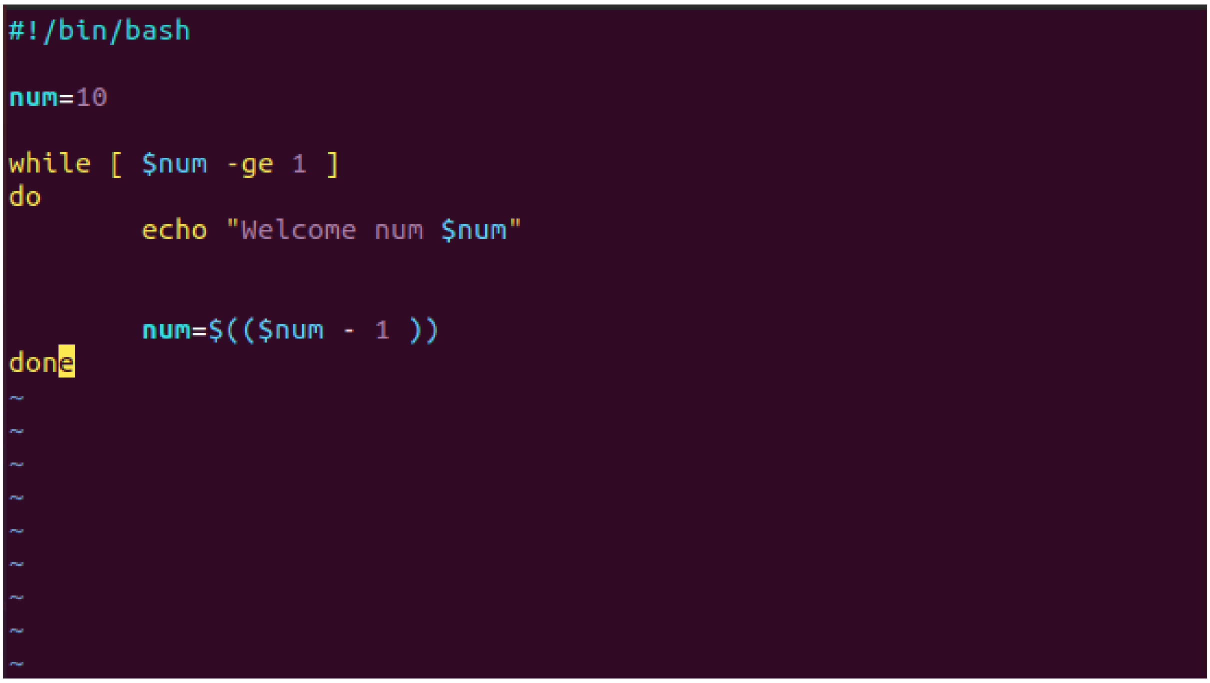
```
#!/bin/bash

echo "Give a num between 1 to 5 "

num=4
while [ $num -ge 0 ]
do
        echo "Welcome num $num"


        num=$(($num - 1 ))
done
~
```

18. Create a while loop which receives an input from the user as the counter and runs that amount of times.



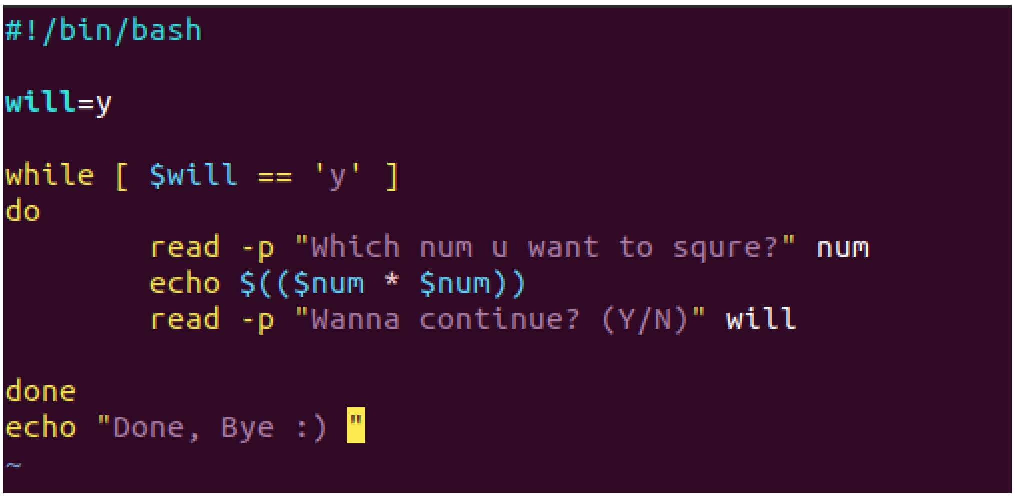
```
#!/bin/bash

echo "Give a num"

read num
while [ $num -ge 1 ]
do
        echo "Welcome num $num"


        num=$(($num - 1 ))
done
~
```

19. Write a Bash script that uses a while loop to print the numbers from 10 down to 1



```
#!/bin/bash

num=10

while [ $num -ge 1 ]
do
        echo "Welcome num $num"


        num=$(($num - 1 ))
done
~
~
~
~
~
~
~
~
~
```

20. Write a Bash script that uses a while loop to prompt the user to enter a number, and then prints the square of that number.



```
#!/bin/bash

will=y

while [ $will == 'y' ]
do
        read -p "Which num u want to squre?" num
        echo $(($num * $num))
        read -p "Wanna continue? (Y/N)" will

done
echo "Done, Bye :) "
~
```

==

GIT:

a) Watch and practice git from the following link:

https://www.youtube.com/watch?v=AzfVDEBn9hw&ab_channel=Simplilearn

b) 1. Create an empty folder named “wpgit”

2. displays the state of the working directory and the staging area using git status

3. Upload your solution of “Weekend Task” in “wpgit”.

4. displays the state of the working directory and the staging area

5. Add the “.git” package folder to your repository using the git “init” command.

6. displays the state of the working directory and the staging area

7. Commit the staged files, using “git commit -m” .

8. displays the state of the working directory and the staging area

9. Show the commit logs.

10. Create a new repository in your GitHub named “weekendtask”.

11. Upload “wpgit” to your repository in github.