


This lab is open book, open notes, open Internet. Be sure to use the *man* logger pages to check the command options.

### Task1:

Please answer the following questions: (20 Point)

1. Install a new ubuntu/fedora VM name it "**finalproject-studentname**"
  - a. Partitions
    - i. 2 CPU
    - ii. 40 GB HDD
    - iii. 2 GB RAM
    - iv. 600 MB /boot
    - v. 1G swap
    - vi. / Rest of the HDD
2. Provide screenshot for the following stages: **System Configuration, Disk configuration and OS update**

Guest OS



Power Status

Powered On

Guest OS

Ubuntu Linux (64-bit)

VMware Tools

Not running, not installed ⓘ

DNS Name

IP Addresses

Encryption

Not encrypted

LAUNCH REMOTE CONSOLE ⓘ

LAUNCH WEB CONSOLE

ACTIONS ▾

⋮

Capacity and Usage

Last updated at 11:53 AM

CPU

1.944 GHz used

2 CPUs allocated

Memory

1.76 GB used

2 GB allocated

Storage

42.08 GB used

42.08 GB allocated

VIEW STATS

VM Hardware

CPU

2 CPU(s), 1992 MHz used

Memory

2 GB, 2 GB memory active

Hard disk 1

40 GB | Thick Provision Lazy Zeroed ⓘ  
CIS215-Datastore-Nimble

Network adapter 1

CIS215Network (connected) | 00:50:56:87:d0:b7

CD/DVD drive 1

Connected 🔌 ▾


Compatibility

ESXi 7.0 U2 and later (VM version 19)


EDIT

Related Objects


Cluster

 CISNutanixCluster


Host


 cisntnx-3.highline.edu

Networks

 CIS215Network

Storage

 CIS215-Datastore-Nimble

 CISBAS-ISO-Store-Nimble

Please answer the following 10 questions (4 points each):

1. The `touch` command can be used to create empty files with a one-line command.
2. The `wall` command is used to broadcast messages to everyone currently logged onto the system.
3. What is `apt-get` used for? **Used to update, remove, and delete software packages.**
4. Write the command line you would use to give the owner of the file `mydata` full rights and read only permission to everyone else. **`sudo chmod 744 mydata`**
5. What command allows you to log in under another user id? **`su`**
6. What command do you use to get input from the user? **`Read`**
7. What command would allow you to determine what programs were eating up the most CPU time? **`Top`**
8. A user on your network has received an error message stating that another user has the same IP address as they do. You need to have the users tell you what their IP address is. What command should they use? **`ifconfig`**
9. What command do you use to print the environmental variables to the screen? **`printenv`**
10. Give an example of how to run a script or command in the background.

```
#!/bin/bash
```

```
while true
```

do

```
echo "Doing Something(My Pid:$$)"
```

sleep 5

done

```
"background.sh" 6L, 77B
```

6,4

ALL

```
student@ubutun22:~$ chmod +x background.sh
student@ubutun22:~$ ./background.sh &
[1] 60990
student@ubutun22:~$ Doing Something(My Pid:60990)
Doing Something(My Pid:60990)
Doing Something(My Pid:60990)
Doing Something(My Pid:60990)
```

```
student@ubutun22:~$ ps -A
```

PID	TTY	TIME	CMD
1	?	00:00:19	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	rcu_gp
4	?	00:00:00	rcu_par_gp
5	?	00:00:00	slub_flushwq
6	?	00:00:00	netns
8	?	00:00:00	kworker/0:0H-events_highpri
10	?	00:00:00	mm_percpu_wq
11	?	00:00:00	rcu_tasks_rude_
12	?	00:00:00	rcu_tasks_trace
13	?	00:00:01	ksoftirqd/0
14	?	00:02:36	rcu_sched
15	?	00:00:04	migration/0
16	?	00:00:00	idle_inject/0
18	?	00:00:00	cpuhp/0
19	?	00:00:00	cpuhp/1
20	?	00:00:00	idle_inject/1
21	?	00:00:04	migration/1
22	?	00:00:00	ksoftirqd/1
24	?	00:00:00	kworker/1:0H-events_highpri
25	?	00:00:00	kdevtmpfs
26	?	00:00:00	inet_frag_wq
27	?	00:00:00	kauditd

```

2367 ?      00:00:00 gsd-disk-utilit
2378 ?      00:00:00 ibus-portal
2381 ?      00:00:00 gsd-printer
2456 ?      00:00:02 ibus-engine-sim
2459 ?      00:00:00 gvfsd-metadata
2477 ?      00:00:00 xdg-desktop-por
2514 ?      00:00:00 gjs
2638 ?      00:00:28 update-notifier
43943 ?     00:00:03 systemd-resolve
43947 ?     00:00:02 systemd-journal
44011 ?     00:00:01 systemd-timesyn
44061 ?     00:07:25 systemd-oomd
44326 ?     00:00:00 systemd-udevd
58737 ?     00:00:00 cupsd
58738 ?     00:00:00 cups-browsed
60547 ?     00:00:01 kworker/0:1-events
60680 ?     00:00:00 kworker/u4:2-flush-8:0
60776 ?     00:00:00 kworker/1:3-events
60827 ?     00:00:00 kworker/u4:1-events_unbound
60902 ?     00:00:00 gjs
60940 ?     00:00:00 kworker/0:0
60941 ?     00:00:00 kworker/u4:3-events_unbound
60955 ?     00:00:00 kworker/1:0-events
60990 ?     00:00:00 background.sh
61039 ?     00:00:00 gnome-terminal-
61057 pts/1  00:00:00 bash
61069 ?     00:00:00 sleep
61070 pts/1  00:00:00 ps

```

```

61070 pts/1  00:00:00 ps
student@ubutun22:~$ kill -9 60990
student@ubutun22:~$ ps -A
  PID TTY          TIME CMD
    1 ?           00:00:19 systemd
    2 ?           00:00:00 kthreadd
    3 ?           00:00:00 rcu_gp
    4 ?           00:00:00 rcu_par_gp
    5 ?           00:00:00 slub_flushwq
    6 ?           00:00:00 netns
    8 ?           00:00:00 kworker/0:0H-events_highpri
   10 ?           00:00:00 mm_percpu_wq
   11 ?           00:00:00 rcu_tasks_rude_
   12 ?           00:00:00 rcu_tasks_trace
   13 ?           00:00:01 ksoftirqd/0
   14 ?           00:02:36 rcu_sched

```

```
student@ubutun22:~$
```

### Task2:

**The next 2 questions are worth 20 points each.**

1. Write a script that tells you today's date current working directory and who is currently logged on.

```
#!/bin/bash
date
whoami
```

1

2



1

"Task2.1.sh" 4L, 26B

4,0-1

All

```
student@ubutun22:~$ bash Task2.1.sh
Tue Mar 14 08:22:24 PM PDT 2023
student
student@ubutun22:~$
```

2. Write a script that:

```
#!/bin/bash  
date  
ls -l  
echo "Have a Great Day"  
  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

"Script1.sh" 4L, 48B      4,23      All

- gives you the time of day. **date**
- gives you a long listing of files in your current directory. **ls -l**
- tells you Have a Great Day at the end. **echo "Have a Great Day"**

Save file=:w Script1.sh



Then Run it= bash Script1.sh

```
student@ubutun22:~$ bash Script1.sh
```

```
Tue Mar 14 05:17:55 PM PDT 2023
```

```
total 84
```

```
drwxrwxr-x 2 student student 4096 Jan 26 14:21 cis-215
drwxr-xr-x 2 student student 4096 Jan 17 15:03 Desktop
drwxr-xr-x 3 student student 4096 Jan 31 14:31 Documents
drwxr-xr-x 3 student student 4096 Jan 17 15:08 Downloads
drwxrwxrwx 2 student student 4096 Feb  9 15:23 foruser1
drwxr-xr-x 2 student student 4096 Jan 17 15:03 Music
drwxrwxr-x 4 student student 4096 Feb  7 15:34 mysamples
drwxrwxr-x 4 student student 4096 Feb  7 15:31 mysamples2
-rw-rw-r-- 1 student student  561 Feb  5 23:54 nanofile1
-rw-rw-r-- 1 student student  307 Feb  5 17:37 nanofile2
drwxr-xr-x 2 student student 4096 Jan 17 15:03 Pictures
drwxr-xr-x 2 student student 4096 Jan 17 15:03 Public
drwxrwxr-x 2 student student 4096 Feb  7 15:16 sample1A
-rw-rw-r-- 1 student student    0 Feb  7 15:26 sample1B
-rw-rw-r-- 1 student student    0 Feb  7 15:15 sample2
-rw-rw-r-- 1 student student  48 Mar 14 17:16 Script1.sh
drwx----- 4 student student 4096 Jan 17 15:08 snap
drwx----- 4 student student 4096 Jan 17 15:08 snap
-rw-rw-r-- 1 student student  137 Jan 25 13:50 'Task 1: Practice Using vim'
drwxr-xr-x 2 student student 4096 Jan 17 15:03 Templates
-rw-rw-r-- 1 student student  137 Jan 31 13:56 test.txt
drwxr-xr-x 2 student student 4096 Jan 17 15:03 Videos
-rw-rw-r-- 1 student student  106 Feb 27 19:18 wfile
-rw-rw-r-- 1 student student  159 Mar  3 15:04 wfiles
-rw-rw-r-- 1 student student    0 Mar  3 15:01 wsampl3
-rw-rw-r-- 1 student student    0 Feb 27 19:10 wsample1
-rw-rw-r-- 1 student student    0 Feb 27 19:10 wsample2
-rw-rw-r-- 1 student student    0 Mar  3 15:02 wsample3
Have a Great Day
```

### Task3 (20 points):

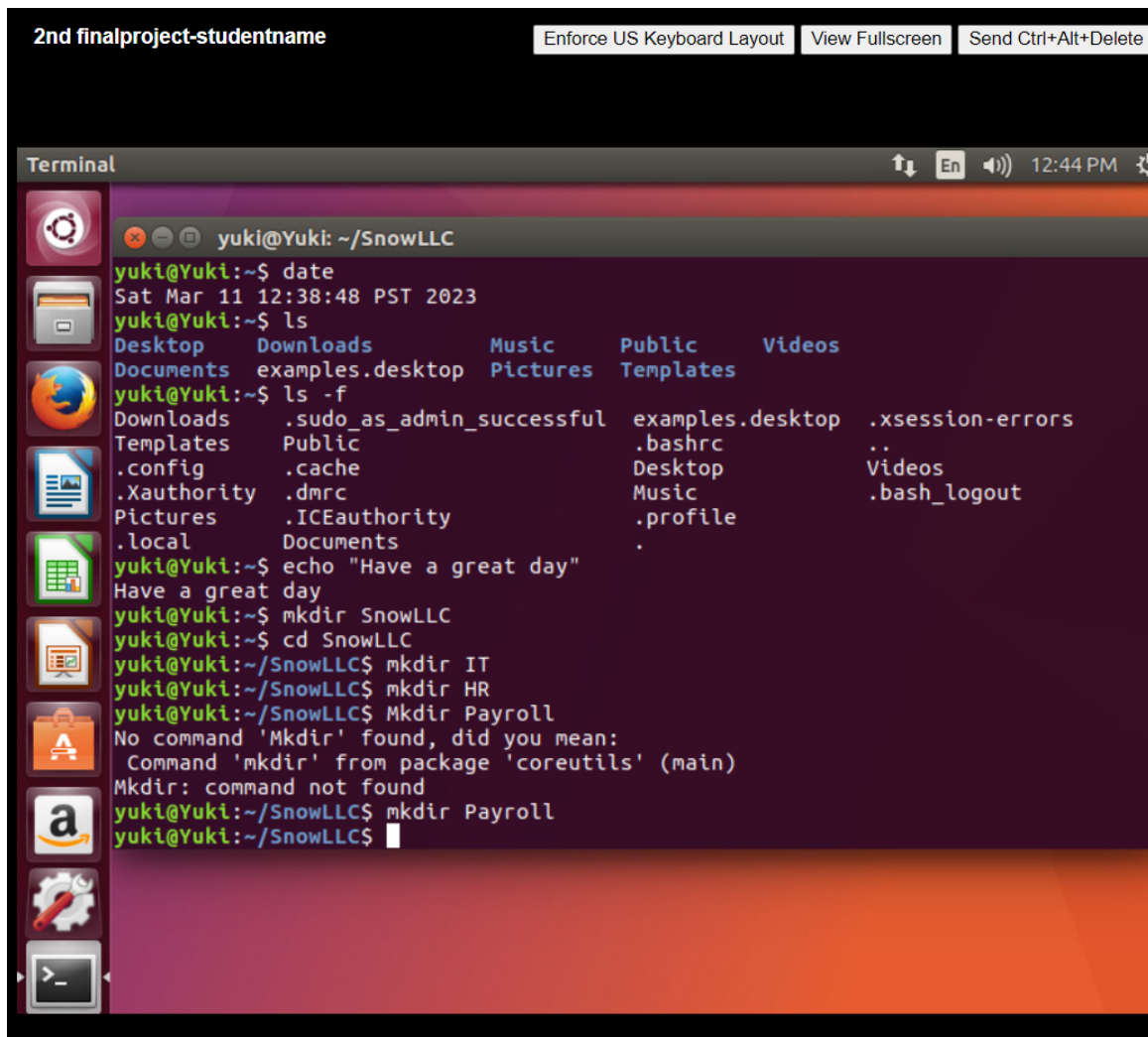
The last 4 questions require you to build the following directory structure.

Main Directory:

- SnowLLC

and 3 subdirectories:

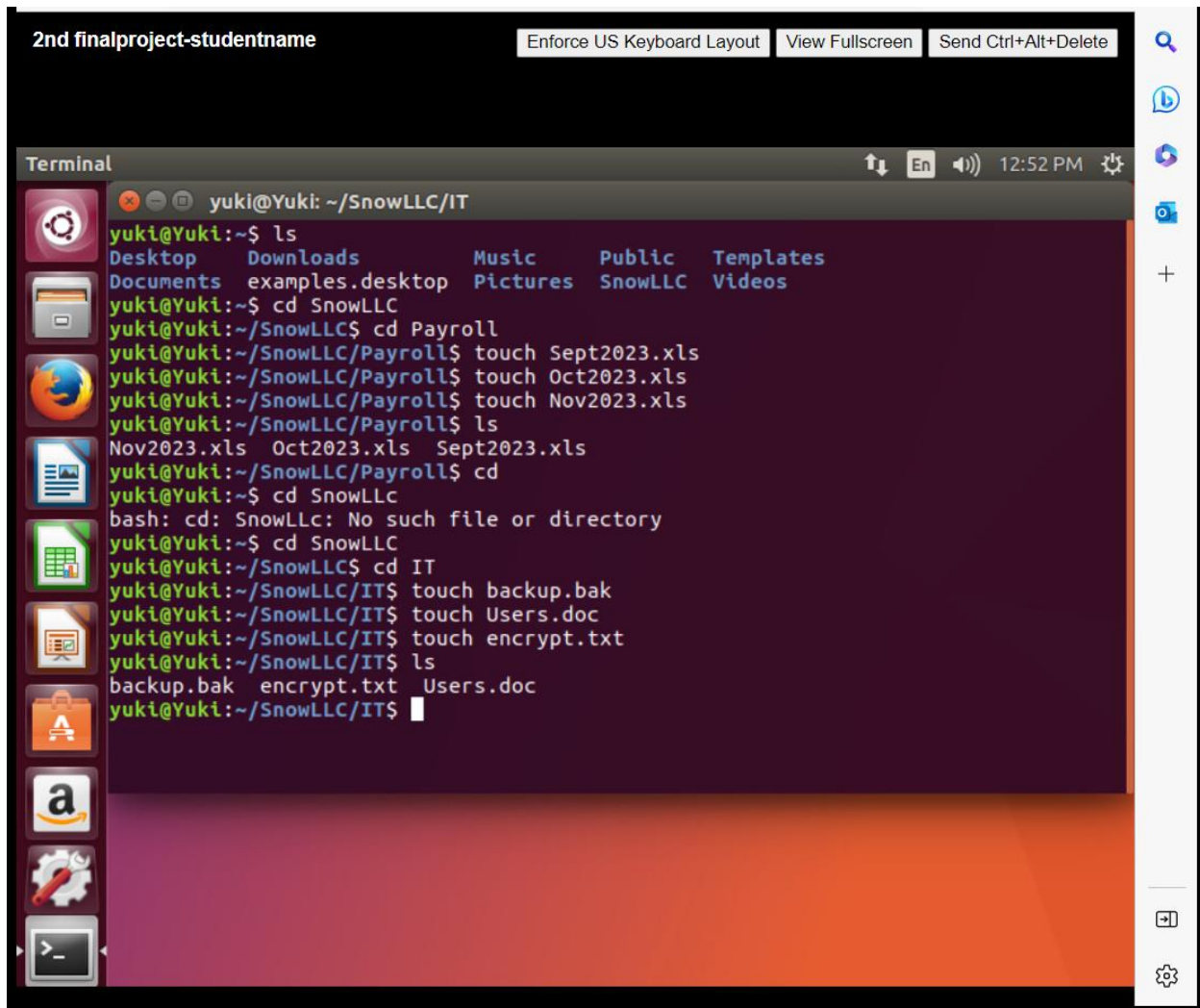
- IT
- HR
- Payroll



The screenshot shows a terminal window titled "2nd finalproject-studentname" with buttons for "Enforce US Keyboard Layout", "View Fullscreen", and "Send Ctrl+Alt+Delete". The terminal output is as follows:

```
yuki@Yuki: ~/SnowLLC
yuki@Yuki:~$ date
Sat Mar 11 12:38:48 PST 2023
yuki@Yuki:~$ ls
Desktop  Downloads      Music  Public  Videos
Documents examples.desktop Pictures  Templates
yuki@Yuki:~$ ls -f
Downloads .sudo_as_admin_successful examples.desktop .xsession-errors
Templates Public .bashrc ..
.config .cache Desktop Videos
.Xauthority .dmrc Music .bash_logout
Pictures .ICEauthority .profile
.local Documents .
yuki@Yuki:~$ echo "Have a great day"
Have a great day
yuki@Yuki:~$ mkdir SnowLLC
yuki@Yuki:~$ cd SnowLLC
yuki@Yuki:~/SnowLLC$ mkdir IT
yuki@Yuki:~/SnowLLC$ mkdir HR
yuki@Yuki:~/SnowLLC$ Mkdir Payroll
No command 'Mkdir' found, did you mean:
Command 'mkdir' from package 'coreutils' (main)
Mkdir: command not found
yuki@Yuki:~/SnowLLC$ mkdir Payroll
yuki@Yuki:~/SnowLLC$
```

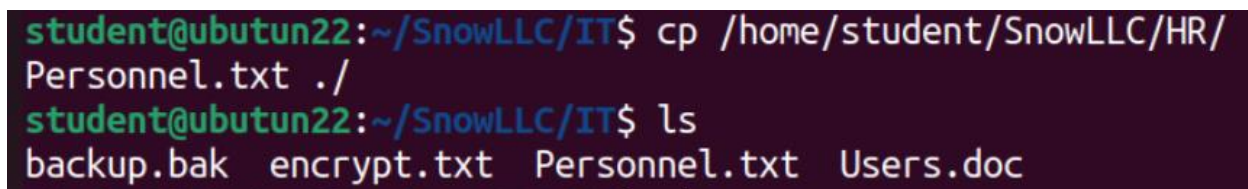
- Under HR, create the files Personnel.txt, Policies.txt, and Employee\_Hbk.txt.
- Under Payroll create the files Sept2023.xls, Oct2023.xls and Nov2023.xls.
- Under IT, create the files backup.bak, Users.doc and encrypt.txt



Attach a screenshot showing that you have completed this task.

Please answer the following questions (5 points each). Each answer must be a command entered on a single command line. Provide screenshots of commands and results.

1. From the IT directory, copy Personnel.txt to your current location.



2. From the Payroll directory, delete Policies.txt.



```
student@ubutun22:~/SnowLLC/Payroll$ rm /home/student/SnowLLC/HR/Policies.txt ./
rm: cannot remove './': Is a directory
student@ubutun22:~/SnowLLC/Payroll$ rm /home/student/SnowLLC/HR/Policies.txt
rm: cannot remove '/home/student/SnowLLC/HR/Policies.txt':
No such file or directory
student@ubutun22:~/SnowLLC/Payroll$ cd
student@ubutun22:~$ cd SnowLLC
student@ubutun22:~/SnowLLC$ cd HR
student@ubutun22:~/SnowLLC/HR$ ls
Employee_Hbk.txt  Personnel.txt
student@ubutun22:~/SnowLLC/HR$
```

3. From SnowLLC level, move all the files under Payroll.

```
student@ubutun22:~/SnowLLC$ mv HR/* IT/* Payroll/
mv: will not overwrite just-created 'Payroll/Personnel.txt'
with 'IT/Personnel.txt'
student@ubutun22:~/SnowLLC$ ls
HR  IT  Payroll
student@ubutun22:~/SnowLLC$ cd
student@ubutun22:~$ cd SnowLLC
student@ubutun22:~/SnowLLC$ cd IT
student@ubutun22:~/SnowLLC/IT$ ls
Personnel.txt
student@ubutun22:~/SnowLLC/IT$ cd ..
student@ubutun22:~/SnowLLC$ cd Payroll
student@ubutun22:~/SnowLLC/Payroll$ ls
backup.bak      Nov2023.xls      Script2
Employee_Hbk.txt Oct2023.xls      Sept2023.xls
encrypt.txt     Personnel.txt     Users.doc
student@ubutun22:~/SnowLLC/Payroll$
```

4. From the SnowLLC level, delete the Payroll directory

```
student@ubutun22:~/SnowLLC$ rm -r Payroll
student@ubutun22:~/SnowLLC$ ls
HR  IT
student@ubutun22:~/SnowLLC$
```

Task4 (20pts each.): (if, while,until & for constructs)

1. Create an if construct that compares whether 2 variables are equal

```
#!/bin/bash
var1=2
var2=2
if [ "$var1" = "$var2" ]; then
    echo "The variable are equal."
else
    echo "The variable are not equal."
fi
~
```

```
student@ubutun22:~/SnowLLC/IT$ vim Script2
student@ubutun22:~/SnowLLC/IT$ bash Script2
The variable are equal.
student@ubutun22:~/SnowLLC/IT$
```

```
#!/bin/bash
var1=2
var2=4
if [ "$var1" = "$var2" ]; then
    echo "The variable are equal."
else
    echo "The variable are not equal."
fi
```

```
student@ubutun22:~/SnowLLC/IT$ vim Script2
student@ubutun22:~/SnowLLC/IT$ bash Script2
The variable are not equal.
student@ubutun22:~/SnowLLC/IT$
```

2. Create a while construct that compares whether 2 variables are equal then output "Great Job Student" if the variable are equal.

```
#!/bin/bash
# Assign two variable
var1="Hello"
var2="Hello"

# Loop while teh variable are not equal
while [ "$var1" != "$var2" ]
do
    # Prompt the user to enter the variable
    echo "Enter two variables: "
    read var1 var2
done

# Output a message if the variables are equal
echo "Great job, student! The variables are equal."

~
~
~
"while.sh" 16L, 326B 15,51 All
```

```
student@ubutun22:~$ bash while.sh
Great job, student! The variables are equal.
student@ubutun22:~$
```

```

#!/bin/bash
# Assign two variable
var1="Hello"
var2="Hell"

# Loop while the variable are not equal
while [ "$var1" != "$var2" ]
do
    # Prompt the user to enter the variable
    echo "Enter two variables: "
    read var1 var2
done

# Output a message if the variables are equal
echo "Great job, student! The variables are equal."

~
~
~
-- INSERT --
4,11 All

```

```

student@ubutun22:~$ bash while.sh
Great job, student! The variables are equal.
student@ubutun22:~$ vim while.sh
student@ubutun22:~$ bash while.sh
Enter two variables:
0
Enter two variables:
1
Enter two variables:
32
Enter two variables:
23
Enter two variables:
0
Enter two variables:
36

```

3. What is the value of counter in the code below after the loop finishes executing?

```
counter = 12
```

```
while counter < 11:
```

```
print(counter)
```

```
counter = counter + 1
```

4. Create a for loop to print numbers from 1 to 10.

```
#!/bin/bash
# Loop through numbers 1 to 10
for i in {1..10}
do
    #Print each number
    echo $i
done
```

```
~
~
~
~
~
~
~
~
~
~
~
~
```

```
"loop.sh" 7L, 98B
```

```
7,4
```

```
All
```



```
student@ubutun22:~$ vim
student@ubutun22:~$ vim loop.sh
student@ubutun22:~$ bash loop.sh
```

```
1
2
3
4
5
6
7
8
9
10
```

```
student@ubutun22:~$
```

**Submission:**

Please submit the answers screenshots of commands and results.