



**Name:** Lujain Zia

**Roll no:** 2023-BSE-034

**Date:** October 31, 2025

## Lab 6

### Task 1 – Switch to root with su - and back to a normal user

**1. Set a root password (Ubuntu root is disabled by default; this enables su - temporarily for the lab):**

```
sudo passwd root# Enter a temporary root password for the lab
```

**Save screenshot as: task1\_set\_root\_password.png**

```
ubuntu@ubuntuserver:~$ sudo passwd root
[sudo] password for ubuntu:
Sorry, try again.
[sudo] password for ubuntu:
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ubuntuserver:~$ |
```

**2. Switch to root and verify:**

```
SU -
whoami
id
```

**Save screenshot as: task1\_su\_root.png**

```
ubuntu@ubuntuserver:~$ su -
Password:
root@ubuntuserver:~# whoami
root
root@ubuntuserver:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntuserver:~#
```

**3. Switch back to your normal user:**

```
exit
whoami
```

**Save screenshot as: task1\_exit\_to\_user.png**

```
root@ubuntuserver:~# exit
logout
ubuntu@ubuntuserver:~$ whoami
ubuntu
ubuntu@ubuntuserver:~$
```

**Task 2 – Create user tom and verify in passwd/group/shadow**

## **1. Create user tom (interactive, sets password and home directory):**

sudo adduser tom

### **Save screenshot as: task2\_adduser\_tom.png**

```
ubuntu@ubuntuserver:~$ sudo adduser tom
info: Adding user 'tom' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'tom' (1001) ...
info: Adding new user 'tom' (1001) with group 'tom (1001)' ...
info: Creating home directory '/home/tom' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for tom
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user 'tom' to supplemental / extra groups 'users' ...
info: Adding user 'tom' to group 'users' ...
ubuntu@ubuntuserver:~$
```

## **2. Verify tom in system files (view and visually confirm presence):**

cat /etc/passwd

### **Save screenshot as: task2\_verify\_passwd.png**

```
libvirt-dnsmasq:x:127:libvirt dnsmasq,,,:/var/run/libvirt-dnsmasq.pid:/bin/sh
tom:x:1001:,:/home/tom:/bin/bash
ubuntu@ubuntuserver:~$ |
```

cat /etc/group

### **Save screenshot as: task2\_verify\_group.png**

```
libvirt-dnsmasq:x:127:
tom:x:1001:
ubuntu@ubuntuserver:~$
```

sudo cat /etc/shadow

### **Save screenshot as: task2\_verify\_shadow.png**

```
libvirt-qemu:x:20386:.....
libvirt-dnsmasq!:20386:::::
tom:$y$j9T$QOBz9CxuSJrbDsMHEr0s1$XELm/hBd8NzMPB6vWhbFRZaE3bRvNk.22W/zex60w0:20393:0:99999:7:::
```

### **Notes:**

**/etc/shadow stores password hashes (not plaintext). You must use sudo to read it.**

## **Task 3 – Create groups; change tom's primary and secondary groups**

**1. Create groups and verify by viewing /etc/group (visually confirm entries exist):**

```
sudo groupadd developer  
sudo groupadd devops  
sudo groupadd designer  
cat /etc/group
```

**Save screenshot as: task3\_groupadd.png**

```
developer:x:1002:  
devops:x:1003:  
designer:x:1004:  
ubuntu@ubuntuserver:~$
```

**2. Change tom's primary group to designer and verify:**

```
sudo usermod -g designer tom  
id tom
```

**Save screenshot as: task3\_change\_primary\_group.png**

```
ubuntu@ubuntuserver:~$ sudo usermod -g designer tom  
ubuntu@ubuntuserver:~$ id  
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),101(lxd),126(libvirt )  
ubuntu@ubuntuserver:~$ id tom  
uid=1001(tom) gid=1004(designer) groups=1004(designer),100(users)  
ubuntu@ubuntuserver:~$
```

**3. Add secondary groups developer and devops to tom and verify:**

```
sudo usermod -aG developer,devops tom  
id tom  
groups tom
```

**Save screenshot as: task3\_add\_secondary\_groups.png**

```
ubuntu@ubuntuserver:~$ sudo usermod -aG developer,devops tom  
ubuntu@ubuntuserver:~$ id tom  
groups tom  
uid=1001(tom) gid=1004(designer) groups=1004(designer),100(users),1002(developer),1003(devops)  
tom : designer users developer devops  
ubuntu@ubuntuserver:~$ |
```

**4. Replace all secondary groups so only tom (user's own group) remains and verify:**

```
sudo usermod -G tom tom  
id tom  
groups tom
```

**Save screenshot as: task3\_reset\_secondary\_groups.png**

```
ubuntu@ubuntuserver:~$ sudo usermod -G tom tom  
id tom  
groups tom  
uid=1001(tom) gid=1004(designer) groups=1004(designer),1001(tom)  
tom : designer tom  
ubuntu@ubuntuserver:~$
```

**Task 4 – Create/delete users (Jerry, Scooby) and groups (jolly, anime)**

**1. Create users:**

```
sudo adduser Jerry
```

```
sudo useradd Scooby
```

**Save screenshot as: task4\_add\_users.png**

```
ubuntu@ubuntuserver:~$ sudo useradd scooby
ubuntu@ubuntuserver:~$ sudo adduser jerry
info: Adding user 'jerry' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'jerry' (1007) ...
info: Adding new user 'jerry' (1007) with group 'jerry (1007)' ...
info: Creating home directory '/home/jerry' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jerry
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
info: Adding new user 'jerry' to supplemental / extra groups 'users' ...
info: Adding user 'jerry' to group 'users' ...
ubuntu@ubuntuserver:~$
```

**2. Try to log in as Scooby immediately (expected authentication failure because there is no password yet):**

```
su - Scooby
```

**Save screenshot as: task4\_scooby\_su\_auth\_failure.png**

```
ubuntu@ubuntuserver:~$ su - scooby
Password:
su: Authentication failure
ubuntu@ubuntuserver:~$ |
```

**3. Set a password for Scooby:**

```
sudo passwd Scooby
```

**Save screenshot as: task4\_set\_password\_scooby.png**

```
ubuntu@ubuntuserver:~$ sudo passwd Scooby
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ubuntuserver:~$
```

**4. Try logging in as Scooby again (home directory still missing; expect a message such as “No directory, logging in with HOME=/”):**

```
su - Scooby
```

**Save screenshot as: task4\_scooby\_su\_no\_home.png**

```
ubuntu@ubuntuserver:~$ su -scooby
Password:
su: failed to execute cooby: No such file or directory
ubuntu@ubuntuserver:~$
```

**5. Show that Scooby’s home directory does not exist yet and what /etc/passwd says:**

```
exit
```

```
cat /etc/passwd
```

```
ls -ld /home/Scooby
```

**Save screenshot as: task4\_scooby\_no\_home.png**

```
Scooby:x:1002:1005::/home/Scooby:/bin/sh
scooby:x:1003:1006::/home/scooby:/bin/sh
jerry:x:1007:1007:,,,:/home/jerry:/bin/bash
ubuntu@ubuntuserver:~$ ls -ld /home/Scooby
ls: cannot access '/home/Scooby': No such file or directory
```

**6. Manually create Scooby's home directory and set proper ownership and permissions:**

```
sudo mkdir -p /home/Scooby
sudo chown Scooby:Scooby /home/Scooby
sudo chmod 750 /home/Scooby
ls -ld /home/Scooby
```

**Save screenshot as: task4\_scooby\_create\_home.png**

```
ubuntu@ubuntuserver:~$ sudo mkdir -p /home/Scooby
sudo chown Scooby:Scooby /home/Scooby
sudo chmod 750 /home/Scooby
ls -ld /home/Scooby
[sudo] password for ubuntu:
drwxr-x--- 2 Scooby Scooby 4096 Nov  1 13:11 /home/Scooby
```

**7. Log in as Scooby again and verify you land in the correct home directory:**

```
su - Scoobypwd
ls -la
```

**Save screenshot as: task4\_scooby\_login\_success.png**

```
ubuntu@ubuntuserver:~$ su - Scooby
Password:
$ su -Scoobypwd
su: invalid option -- 'S'
Try 'su --help' for more information.
$ ls -la
total 8
drwxr-x--- 2 Scooby Scooby 4096 Nov  1 13:11 .
drwxr-xr-x  7 root    root   4096 Nov  1 13:12 ..
$ |
```

**8. Verify users in system files and observe shell of Scooby:**

```
exit
cat /etc/passwd
```

**Save screenshot as: task4\_verify\_users.png**

```
tom:x:1001:1004:,,,:/home/tom:/bin/bash
Scooby:x:1002:1005::/home/Scooby:/bin/sh
scooby:x:1003:1006::/home/scooby:/bin/sh
jerry:x:1007:1007:,,,:/home/jerry:/bin/bash
ubuntu@ubuntuserver:~$
```

**9. Change the shell from /bin/sh to /bin/bash**

```
sudo usermod -s /bin/bash Scooby
su - Scooby
```

**Save screenshot as: task4\_shell\_switching.png**

```
ubuntu@ubuntuserver:~$ sudo usermod -s /bin/bash Scooby
su - Scooby
Password:
Scooby@ubuntuserver:~$ exit
logout
tom:x:1001:1004:,:/home/tom:/bin/bash
Scooby:x:1002:1005:,:/home/Scooby:/bin/bash
scooby:x:1003:1006:,:/home/scooby:/bin/sh
jerry:x:1007:1007:,:/home/jerry:/bin/bash
ubuntu@ubuntuserver:~$ |
```

## 10. Create groups:

```
sudo addgroup jolly
sudo groupadd anime
```

**Save screenshot as: task4\_add\_groups.png**

```
ubuntu@ubuntuserver:~$ sudo addgroup jolly
sudo groupadd anime
info: Selecting GID from range 1000 to 59999 ...
info: Adding group 'jolly' (GID 1008) ...
ubuntu@ubuntuserver:~$ |
```

## 11. Verify groups:

```
cat /etc/group
```

**Save screenshot as: task4\_verify\_groups.png**

```
jerry:x:1007:
jolly:x:1008:
anime:x:1009:
ubuntu@ubuntuserver:~$ |
```

## 12. Delete groups and users:

```
sudo delgroup jolly
sudo groupdel anime
cat /etc/group
```

```
sudo deluser --remove-home Jerry
sudo userdel -r Scooby
cat /etc/passwd
```

**Save screenshots**

**as: task4\_delete\_groups.png, task4\_delete\_users.png**

**Deleting groups:**

```
ubuntu@ubuntuserver:~$ sudo delgroup jolly
sudo groupdel anime
cat /etc/group
info: Removing group 'jolly' ...
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,ubuntu
ubuntu:x:1000:1000:Ubuntu user,,,:/home/ubuntu:/bin/bash
ubuntu@ubuntuserver:~$ |
```

```
designer:x:1004:  
Scooby:x:1005:  
scooby:x:1006:  
jerry:x:1007:  
ubuntu@ubuntuserver:~$
```

### **Deleting users:**

```
ubuntu@ubuntuserver:~$ sudo deluser --remove-home jerry  
sudo userdel -r Scooby  
cat /etc/passwd  
info: Looking for files to backup/remove ...  
info: Removing files ...  
info: Removing crontab ...  
info: Removing user 'jerry' ...  
userdel: user 'Scooby' does not exist  
libvirt-qemu:x:64055:994:Libvirt Qemu,,,:/var/lib/libvirt  
libvirt-dnsmasq:x:121:127:Libvirt Dnsmasq,,,:/var/lib/l:  
tom:x:1001:1004:,,,:/home/tom:/bin/bash
```

## **Task 5 – Create user Student; create files; set owner/group; identify file types**

### **1. Create Student:**

sudo adduser Student

#### **Save screenshot as: task5\_create\_student.png**

```
ubuntu@ubuntuserver:~$ sudo adduser student  
info: Adding user 'student' ...  
info: Selecting UID/GID from range 1000 to 59999 ...  
info: Adding new group 'student' (1005) ...  
info: Adding new user 'student' (1005) with group 'student (1005)' ...  
info: Creating home directory '/home/student' ...  
info: Copying files from '/etc/skel' ...  
New password:  
Retype new password:  
Sorry, passwords do not match.  
passwd: Authentication token manipulation error  
passwd: password unchanged  
Try again? [y/N] y  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for student  
Enter the new value, or press ENTER for the default  
    Full Name []:  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Is the information correct? [Y/n] y  
info: Adding new user 'student' to supplemental / extra groups 'users' ...  
info: Adding user 'student' to group 'users' ...  
ubuntu@ubuntuserver:~$
```

### **2. Switch to Student and create files:**

su - Student

touch file1

mkdir -p dir1

touch dir1/file2

ls -l

**Save screenshot as: task5\_create\_files.png**

```
ubuntu@ubuntuserver:~$ su - student
Password:
student@ubuntuserver:~$ touch file1
mkdir -p dir1
touch dir1/file2
student@ubuntuserver:~$ ls -l
total 4
drwxrwxr-x 2 student student 4096 Nov  1 13:32 dir1
-rw-rw-r-- 1 student student     0 Nov  1 13:32 file1
student@ubuntuserver:~$ cd dir1
student@ubuntuserver:~/dir1$ ls -l
total 0
-rw-rw-r-- 1 student student 0 Nov  1 13:32 file2
student@ubuntuserver:~/dir1$
```

**3. Change owner then group for file1 (separate commands):**

sudo chown tom file1

ls -l file1

sudo chgrp devops file1

ls -l file1

**Save screenshots**

**as: task5\_chown\_file1.png, task5\_chgrp\_file1.png**

```
student@ubuntuserver:~$ sudo chown tom file1
[sudo] password for student:
student@ubuntuserver:~$ ls -l file1
-rw-rw-r-- 1 tom student 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
student@ubuntuserver:~$ sudo chgrp devops file1
student@ubuntuserver:~$ ls -l file1
-rw-rw-r-- 1 tom devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

**4. Identify files/directories and show /dev/null:**

ls -l

ls -l dir1

ls -l /dev/null

file file1 dir1 /dev/null

**Save screenshot as: task5\_file\_types.png**

```
student@ubuntuserver:~$ ls -l
ls -l dir1
total 4
drwxrwxr-x 2 student student 4096 Nov  1 13:32 dir1
-rw-rw-r-- 1 tom      devops     0 Nov  1 13:32 file1
total 0
-rw-rw-r-- 1 student student 0 Nov  1 13:32 file2
student@ubuntuserver:~$ ls -l /dev/null
file file1 dir1 /dev/null
crw-rw-rw- 1 root   root 1, 3 Nov  1 12:00 /dev/null
file1:    empty
dir1:    directory
/dev/null: character special (1/3)
student@ubuntuserver:~$ |
```

## 5. Exit Student:

```
exit
```

**Save screenshot as: task5\_exit\_student.png**

```
student@ubuntuserver:~$ exit
logout
ubuntu@ubuntuserver:~$ |
```

## Task 6 – Change permissions using symbolic mode

**Target file: ~/file1 (run these as the Student user)**

### 1. Ensure Student and file present:

```
su - Studentcd ~
ls -l file1
```

**Save screenshot as: task6\_su\_student.png**

```
student@ubuntuserver:~$ ls -l file1
-rw-rw-r-- 1 tom devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

### 2. Remove all permissions:

```
chmod -rwx file1
ls -l file1
```

**Save screenshot as: task6\_chmod\_remove\_rwx.png**

```
student@ubuntuserver:~$ chmod -rwx file1
student@ubuntuserver:~$ ls -l file1
----- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

### 3. Add read to all:

```
chmod +r file1
ls -l file1
```

**Save screenshot as: task6\_chmod\_add\_r.png**

```
student@ubuntuserver:~$ chmod +r file1
ls -l file1
-r--r--r-- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

**4. Add execute to user:**

```
chmod u+x file1  
ls -l file1
```

**Save screenshot as: task6\_chmod\_u\_plus\_x.png**

```
student@ubuntuserver:~$ chmod u+x file1  
ls -l file1  
-r-xr--r-- 1 student devops 0 Nov  1 13:32 file1  
student@ubuntuserver:~$ |
```

**5. Add write to user and group:**

```
chmod ug+w file1  
ls -l file1
```

**Save screenshot as: task6\_chmod\_ug\_plus\_w.png**

```
student@ubuntuserver:~$ chmod ug+w file1  
ls -l file1  
-rwxrw-r-- 1 student devops 0 Nov  1 13:32 file1  
student@ubuntuserver:~$ |
```

**6. Remove all permissions (explicit):**

```
chmod ugo-rwx file1  
ls -l file1
```

**Save screenshot as: task6\_chmod\_ugo\_minus\_rwx.png**

```
student@ubuntuserver:~$ chmod ugo-rwx file1  
ls -l file1  
----- 1 student devops 0 Nov  1 13:32 file1  
student@ubuntuserver:~$ |
```

**Task 7 – Change permissions using “set” symbolic form (u= g= o=)****1. Ensure you are Student:**

```
su - Studentcd ~  
ls -l file1
```

**Save screenshot as: task7\_student\_context.png**

```
student@ubuntuserver:~$ ls -l file1  
----- 1 student devops 0 Nov  1 13:32 file1  
student@ubuntuserver:~$ |
```

**2. Set all to rwx:**

```
chmod u=rwx,g=rwx,o=rwx file1  
ls -l file1
```

**Save screenshot as: task7\_chmod\_set\_all\_rwx.png**

```
student@ubuntuserver:~$ chmod u=rwx,g=rwx,o=rwx file1  
ls -l file1  
-rwxrwxrwx 1 student devops 0 Nov  1 13:32 file1  
student@ubuntuserver:~$ |
```

**3. Remove execute from group and others:**

```
chmod g=rw,o=rw file1  
ls -l file1
```

**Save screenshot as: task7\_remove\_exec\_go.png**

```
student@ubuntuserver:~$ chmod g=rw,o=rw file1
ls -l file1
-rwxrw-rw- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

#### 4. Remove all permissions:

```
chmod u,g,o= file1
ls -l file1
```

**Save screenshot as: task7\_remove\_all\_perms.png**

```
student@ubuntuserver:~$ chmod u,g,o= file1
ls -l file1
----- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

### Task 8 – Change permissions using numeric (octal) mode

#### 1. Ensure you are Student:

```
su - Studentcd ~
ls -l file1
```

**Save screenshot as: task8\_student\_context.png**

```
student@ubuntuserver:~$ ls -l file1
----- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

#### 2. Run each command and capture screenshot after each ls:

```
chmod 777 file1
ls -l file1
```

**task8\_chmod\_777.png**

```
student@ubuntuserver:~$ chmod 777 file1
ls -l file1
-rwxrwxrwx 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

#### 3.

```
chmod 700 file1
ls -l file1
```

**task8\_chmod\_700.png**

```
student@ubuntuserver:~$ chmod 700 file1
ls -l file1
-rwx----- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

#### 4.

```
chmod 744 file1
ls -l file1
```

**task8\_chmod\_744.png**

```
student@ubuntuserver:~$ chmod 744 file1
ls -l file1
-rwxr--r-- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

#### 5.

```
chmod 640 file1
```

```
ls -l file1
```

### **task8\_chmod\_640.png**

```
student@ubuntuserver:~$ chmod 640 file1
ls -l file1
-rw-r----- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

**6.**

```
chmod 664 file1
```

```
ls -l file1
```

### **task8\_chmod\_664.png**

```
student@ubuntuserver:~$ chmod 664 file1
ls -l file1
-rw-rw-r-- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

**7.**

```
chmod 775 file1
```

```
ls -l file1
```

### **task8\_chmod\_775.png**

```
student@ubuntuserver:~$ chmod 775 file1
ls -l file1
-rwxrwxr-x 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

**8.**

```
chmod 750 file1
```

```
ls -l file1
```

### **task8\_chmod\_750.png**

```
student@ubuntuserver:~$ chmod 750 file1
ls -l file1
-rwxr-x--- 1 student devops 0 Nov  1 13:32 file1
student@ubuntuserver:~$ |
```

## **Task 9 – Practice pipes, pagers, grep, and redirects with /var/log/syslog**

### **1. less:**

```
sudo cat /var/log/syslog | less # quit q
```

### **task9\_grep\_less.png**

```
student@ubuntuserver: ~      x  +  v
2025-11-01T12:00:43.902345+00:00 ubuntuserver rsyslogd: [origin software="rsyslogd" sw
info="https://www.rsyslog.com"] rsyslogd was HUPed
2025-11-01T12:00:44.027034+00:00 ubuntuserver systemd[1]: logrotate.service: Deactivat
2025-11-01T12:00:44.027322+00:00 ubuntuserver systemd[1]: Finished logrotate.service -
2025-11-01T12:00:44.027769+00:00 ubuntuserver systemd[1]: logrotate.service: Consumed
    0B memory swap peak.
2025-11-01T12:00:44.058586+00:00 ubuntuserver xrdp[1513]: [INFO ] starting xrdp with p
2025-11-01T12:00:44.064689+00:00 ubuntuserver xrdp[1513]: [INFO ] address [0.0.0.0] po
2025-11-01T12:00:44.066358+00:00 ubuntuserver xrdp[1513]: [INFO ] listening to port 33
2025-11-01T12:00:44.071323+00:00 ubuntuserver xrdp[1513]: [INFO ] xrdp_listen_pp done
2025-11-01T12:00:44.225606+00:00 ubuntuserver snapd[1100]: overlord.go:288: Acquiring
2025-11-01T12:00:44.225844+00:00 ubuntuserver snapd[1100]: overlord.go:293: Acquired s
2025-11-01T12:00:44.302836+00:00 ubuntuserver snapd[1100]: daemon.go:276: started snap
/24.04 (amd64) linux/6.8.0-86-generic.
2025-11-01T12:00:44.458441+00:00 ubuntuserver kernel: loop3: detected capacity change
2025-11-01T12:00:44.505801+00:00 ubuntuserver snapd[1100]: daemon.go:370: adjusting st
estimate of 30s plus 5s per snap)
2025-11-01T12:00:44.512182+00:00 ubuntuserver systemd[1]: tmp-syscheck\x2dmountpoint\x
ccessfully.
2025-11-01T12:00:44.620164+00:00 ubuntuserver snapd[1100]: backends.go:70: AppArmor st
features are available (using snapd provided apparmor_parser)
2025-11-01T12:00:44.638364+00:00 ubuntuserver systemd[1]: Finished apport.service - au
2025-11-01T12:00:44.737250+00:00 ubuntuserver systemd[1]: dmesg.service: Deactivated s
2025-11-01T12:00:44.925366+00:00 ubuntuserver snapd[1100]: backend.go:141: delegating
    system snap security profiles setup
2025-11-01T12:00:44.972518+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00
d config version '1' has been deprecated and will be converted on each startup in cont
g migrate` after upgrading to containerd 2.0 to avoid conversion on startup"
2025-11-01T12:00:44.972954+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00
":
```

## 2. more:

```
sudo cat /var/log/syslog | more
```

### task9\_grep\_more.png

```
2025-11-01T12:00:43.902345+00:00 ubuntuserver rsyslogd: [origin software="rsyslogd" sw
info="https://www.rsyslog.com"] rsyslogd was HUPed
2025-11-01T12:00:44.027034+00:00 ubuntuserver systemd[1]: logrotate.service: Deactivat
2025-11-01T12:00:44.027322+00:00 ubuntuserver systemd[1]: Finished logrotate.service -
2025-11-01T12:00:44.027769+00:00 ubuntuserver systemd[1]: logrotate.service: Consumed
    0B memory swap peak.
2025-11-01T12:00:44.058586+00:00 ubuntuserver xrdp[1513]: [INFO ] starting xrdp with p
2025-11-01T12:00:44.064689+00:00 ubuntuserver xrdp[1513]: [INFO ] address [0.0.0.0] po
2025-11-01T12:00:44.066358+00:00 ubuntuserver xrdp[1513]: [INFO ] listening to port 33
2025-11-01T12:00:44.071323+00:00 ubuntuserver xrdp[1513]: [INFO ] xrdp_listen_pp done
2025-11-01T12:00:44.225606+00:00 ubuntuserver snapd[1100]: overlord.go:288: Acquiring
2025-11-01T12:00:44.225844+00:00 ubuntuserver snapd[1100]: overlord.go:293: Acquired s
2025-11-01T12:00:44.302836+00:00 ubuntuserver snapd[1100]: daemon.go:276: started snap
/24.04 (amd64) linux/6.8.0-86-generic.
2025-11-01T12:00:44.458441+00:00 ubuntuserver kernel: loop3: detected capacity change
2025-11-01T12:00:44.505801+00:00 ubuntuserver snapd[1100]: daemon.go:370: adjusting st
estimate of 30s plus 5s per snap)
2025-11-01T12:00:44.512182+00:00 ubuntuserver systemd[1]: tmp-syscheck\x2dmountpoint\x
ccessfully.
2025-11-01T12:00:44.620164+00:00 ubuntuserver snapd[1100]: backends.go:70: AppArmor st
features are available (using snapd provided apparmor_parser)
2025-11-01T12:00:44.638364+00:00 ubuntuserver systemd[1]: Finished apport.service - au
2025-11-01T12:00:44.737250+00:00 ubuntuserver systemd[1]: dmesg.service: Deactivated s
2025-11-01T12:00:44.925366+00:00 ubuntuserver snapd[1100]: backend.go:141: delegating
    system snap security profiles setup
2025-11-01T12:00:44.972518+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00
d config version '1' has been deprecated and will be converted on each startup in cont
g migrate` after upgrading to containerd 2.0 to avoid conversion on startup"
2025-11-01T12:00:44.972954+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00
"--More--
```

## 3. grep failures/errors:

```
sudo grep -E 'fail|error' /var/log/syslog | head
```

### task9\_grep\_head.png

```

student@ubuntuserver:~$ sudo grep -E 'fail|error' /var/log/syslog | head
2025-11-01T12:00:45.108493+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.107789122Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.aufs\"..." error="aufs is not supported (modprobe aufs failed: exit sta
tus 1 \\"modprobe: FATAL: Module aufs not found in directory /lib/modules/6.8.0-86-generic\\n\\"): skip plugin" type=io.co
ntainerd.snapshotter.v1
2025-11-01T12:00:45.114223+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113006059Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.blockfile\"..." error="no scratch file generator: skip plugin" type=io.
containerd.snapshotter.v1
2025-11-01T12:00:45.114521+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113611499Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.btrfs\"..." error="path /var/lib/containerd/io.containerd.snapshotter.v
1.btrfs (ext4) must be a btrfs filesystem to be used with the btrfs snapshotter: skip plugin" type=io.containerd.snapshot
ter.v1
2025-11-01T12:00:45.114620+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113667991Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.devmapper\"..." error="devmapper not configured: skip plugin" type=io.c
ontainerd.snapshotter.v1
2025-11-01T12:00:45.117241+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.116617183Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.zfs\"..." error="path /var/lib/containerd/io.containerd.snapshotter.v1.
zfs must be a zfs filesystem to be used with the zfs snapshotter: skip plugin" type=io.containerd.snapshotter.v1
2025-11-01T12:00:45.157755+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.152569411Z" level=info msg="sk
ip loading plugin \"io.containerd.tracing.processor.v1.otlp\"..." error="skip plugin: tracing endpoint not configured" t
ype=io.containerd.tracing.processor.v1
2025-11-01T12:00:45.158436+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.152712262Z" level=info msg="sk
ip loading plugin \"io.containerd.internal.v1.tracing\"..." error="skip plugin: tracing endpoint not configured" type=io
.containerd.internal.v1
2025-11-01T12:00:47.359789+00:00 ubuntuserver dockerd[1666]: time="2025-11-01T12:00:47.358707171Z" level=info msg="CDI d
irectory does not exist, skipping: failed to monitor for changes: no such file or directory" dirs=/etc/cdi
2025-11-01T12:00:47.359883+00:00 ubuntuserver dockerd[1666]: time="2025-11-01T12:00:47.358787084Z" level=info msg="CDI d
irectory does not exist, skipping: failed to monitor for changes: no such file or directory" dirs=/var/run/cdi
2025-11-01T12:00:48.272164+00:00 ubuntuserver multipath: sda: failed to get sysfs uid: No such file or directory
student@ubuntuserver:~$ "

```

#### 4. redirect:

sudo grep -i systemd /var/log/syslog > ~/syslog\_systemd.txt

#### task9\_redirect\_overwrite.png

```

student@ubuntuserver:~$ sudo grep -i systemd /var/log/syslog > ~/syslog_systemd.txt
student@ubuntuserver:~$ "

```

#### append:

sudo grep -i network /var/log/syslog >> ~/syslog\_systemd.txt

cat ~/syslog\_systemd.txt

#### task9\_redirect\_append.png

```

student@ubuntuserver:~$ 
2025-11-01T12:59:12.218723+00:00 ubuntuserver systemd-timesyncd[726]: Network configuration changed, trying to establish
connection.
2025-11-01T13:23:01.712224+00:00 ubuntuserver systemd-networkd[712]: ens33: Lost carrier
2025-11-01T13:23:01.712724+00:00 ubuntuserver systemd-networkd[712]: ens33: DHCP lease lost
2025-11-01T13:23:01.737411+00:00 ubuntuserver systemd-networkd[712]: ens33: DHCPv6 lease lost
2025-11-01T13:23:02.059794+00:00 ubuntuserver systemd-timesyncd[726]: No network connectivity, watching for changes.
2025-11-01T13:23:07.823721+00:00 ubuntuserver systemd-networkd[712]: ens33: Gained carrier
2025-11-01T13:23:07.828714+00:00 ubuntuserver NetworkManager[1144]: <info> [1762003387.8252] device (ens33): carrier: l
ink connected
2025-11-01T13:23:07.884257+00:00 ubuntuserver systemd-timesyncd[726]: Network configuration changed, trying to establish
connection.
2025-11-01T13:23:07.893435+00:00 ubuntuserver systemd-networkd[712]: ens33: DHCPv4 address 192.168.85.133/24, gateway 19
2.168.85.2 acquired from 192.168.85.254
2025-11-01T13:23:07.915877+00:00 ubuntuserver systemd-timesyncd[726]: Network configuration changed, trying to establish
connection.
2025-11-01T13:31:50.104062+00:00 ubuntuserver systemd-networkd[712]: ens33: Lost carrier
2025-11-01T13:31:50.104431+00:00 ubuntuserver systemd-networkd[712]: ens33: DHCP lease lost
2025-11-01T13:31:50.1109423+00:00 ubuntuserver systemd-networkd[712]: ens33: DHCPv6 lease lost
2025-11-01T13:31:54.276351+00:00 ubuntuserver NetworkManager[1144]: <info> [1762003914.2431] device (ens33): carrier: l
ink connected
2025-11-01T13:31:54.288697+00:00 ubuntuserver systemd-networkd[712]: ens33: Gained carrier
2025-11-01T13:31:54.391115+00:00 ubuntuserver systemd-timesyncd[726]: Network configuration changed, trying to establish
connection.
2025-11-01T13:31:54.396596+00:00 ubuntuserver systemd-networkd[712]: ens33: DHCPv4 address 192.168.85.133/24, gateway 19
2.168.85.2 acquired from 192.168.85.254
2025-11-01T13:31:54.405181+00:00 ubuntuserver systemd-timesyncd[726]: Network configuration changed, trying to establish
connection.
2025-11-01T13:46:54.230741+00:00 ubuntuserver systemd-timesyncd[726]: Network configuration changed, trying to establish
connection.
student@ubuntuserver:~$ "

```

#### Alternative (journalctl) if needed:

sudo journalctl | less

sudo journalctl -u systemd | grep -i error > ~/journal\_errors.txt

#### task9\_journalctl\_alternative.png

```
student@ubntuserver:~ + - X Oct 24 09:37:45 ubntuserver kernel: Linux version 6.8.0-86-generic (buildd@lcy02-amd64-031) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0, GNU ld (GNU Binutils for Ubuntu) 2.42) #87-Ubuntu SMP PREEMPT_DYNAMIC Mon Sep 22 18:03:36 UTC 2025 (Ubuntu 6.8.0-86.87-generic 6.8.12) Oct 24 09:37:45 ubntuserver kernel: Command line: BOOT_IMAGE=/vmlinuz-6.8.0-86-generic root=/dev/mapper/ubuntu--vg-ubuntu--lv_ro Oct 24 09:37:45 ubntuserver kernel: KERNEL supported cpus: Oct 24 09:37:45 ubntuserver kernel: Intel GenuineIntel Oct 24 09:37:45 ubntuserver kernel: AMD AuthenticAMD Oct 24 09:37:45 ubntuserver kernel: Hygon HygonGenuine Oct 24 09:37:45 ubntuserver kernel: Centaur CentaurHauls Oct 24 09:37:45 ubntuserver kernel: zhaoxin Shanghai Oct 24 09:37:45 ubntuserver kernel: Disabled fast string operations Oct 24 09:37:45 ubntuserver kernel: BIOS-provided physical RAM map: Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000000e7fff] usable Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x00000000000e800-0x000000000009ffff] reserved Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x0000000000dc000-0x0000000000ffff] reserved Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000bfecffff] usable Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x00000000bfed0000-0x00000000bfefefff] ACPI data Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x00000000bfef0000-0x00000000bfeffffff] ACPI NVS Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x00000000bf000000-0x00000000bfffffff] usable Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x000000000fec0000-0x000000000fec0ffff] reserved Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x000000000fee0000-0x000000000fee0ffff] reserved Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x000000000ffe0000-0x000000000ffffffff] reserved Oct 24 09:37:45 ubntuserver kernel: BIOS-e820: [mem 0x0000000010000000-0x000000013fffffff] usable Oct 24 09:37:45 ubntuserver kernel: NX (Execute Disable) protection: active Oct 24 09:37:45 ubntuserver kernel: APIC: Static calls initialized Oct 24 09:37:45 ubntuserver kernel: SMBIOS 2.7 present. Oct 24 09:37:45 ubntuserver kernel: DMI: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6. :|
```

## **Task 10 – Script setup.sh – variables, command substitution, file/dir checks, permissions (use vim)**

## **1. Include bash shebang**

**Code to add (enter in vim as the first line of the file):**

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

## **Steps:**

vim setup.sh → add the shebang line → save and quit

```
chmod +x setup.sh
```

`./setup.sh`

### Screenshots:

**vim editor showing the shebang: task10\_b1\_vim.png**

**script run output (likely no output but show ./setup.sh)**

run): task19 b1 run.png

```
ubuntu@ubuntuserver:~$ vim setup.sh
ubuntu@ubuntuserver:~$ |
ubuntu@ubuntuserver:~$ chmod +x setup.sh
./setup.sh
ubuntu@ubuntuserver:~$ |
```

## 2. Define variable var1 and echo it

### Code to append:

```
# Define and show var1
var1="Hello from Lab 6"echo "var1: $var1"
```

### Steps:

vim setup.sh → append the code above → save and quit  
./setup.sh

### Screenshots:

**vim editor showing var1 code appended: task10\_b2\_vim.png**

**script run output showing var1 printed: task10\_b2\_run.png**

The screenshot shows a terminal window with a vim session. The vim buffer contains the following code:

```
#!/bin/bash
var1="Hello from Lab 6"
echo "var1: $var1"
#
```

Below the vim window, the terminal command line shows the output of running the script:

```
ubuntu@ubuntuserver:~$ ./setup.sh
var1: Hello from Lab 6
ubuntu@ubuntuserver:~$ |
```

## 3. Save output of ls -l into variable allFiles and echo it

### Code to append:

```
# Save ls -l to variable and display
allFiles=$(ls -l)"echo "allFiles $(ls -l);echo "$allFiles"
```

### Steps:

vim setup.sh → append the code above → save and quit  
./setup.sh

### Screenshots:

**vim editor showing allFiles code appended: task10\_b3\_vim.png**

## **script run output showing the ls -l content**

**echoed: task10\_b3\_run.png**

**4. If directory dir1 exists echo a message; else create it**

## **Code to append:**

```
# Directory checkif [ -d "dir1" ]; then  
echo "Directory dir1 exists."else  
echo "Directory dir1 does not exist. Creating..."  
mkdir -p "dir1"  
echo "Directory dir1 created."fi
```

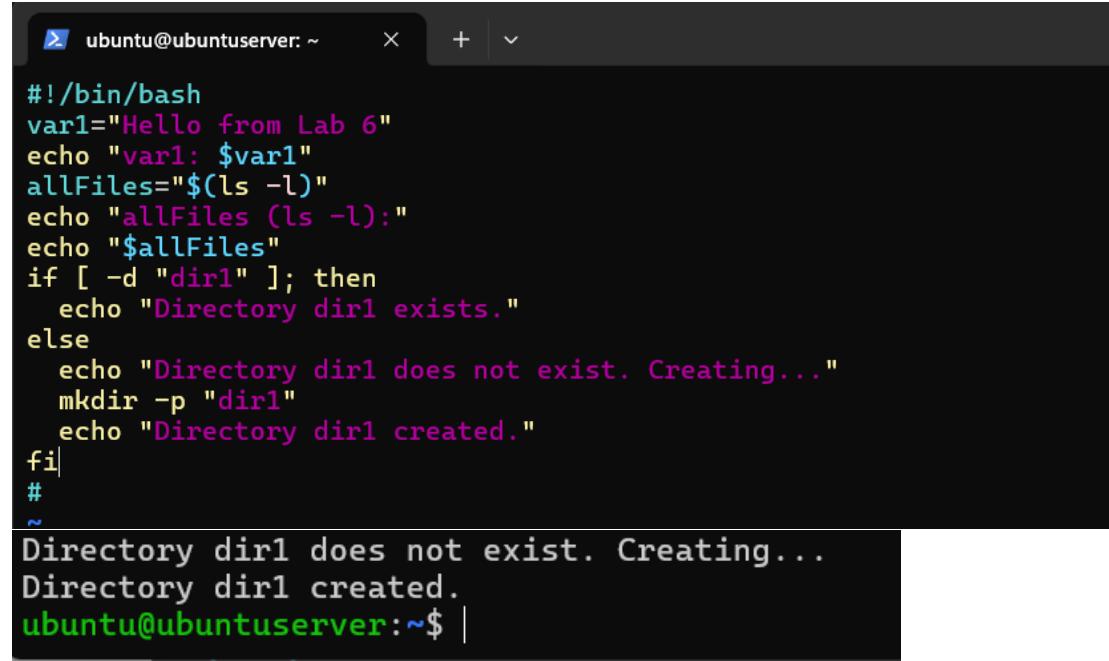
## **Steps:**

vim setup.sh → append the code above → save and quit  
./setup.sh

### Screenshots:

vim editor showing dir1 check code: task10\_b4 vim.png

**script run output showing directory message or creation: task10\_b4\_run.png**



A terminal window titled "ubuntu@ubuntuserver: ~" showing the execution of a bash script. The script checks for a variable var1, lists all files in the current directory, and creates a directory dir1 if it does not exist. The output shows the directory was created.

```
#!/bin/bash
var1="Hello from Lab 6"
echo "var1: $var1"
allFiles="$(ls -l)"
echo "allFiles (ls -l):"
echo "$allFiles"
if [ -d "dir1" ]; then
    echo "Directory dir1 exists."
else
    echo "Directory dir1 does not exist. Creating..."
    mkdir -p "dir1"
    echo "Directory dir1 created."
fi
#
~
Directory dir1 does not exist. Creating...
Directory dir1 created.
ubuntu@ubuntuserver:~$ |
```

**5. If file dir1/file2 does not exist, create it**

**Code to append:**

```
# File checkif [ -f "dir1/file2" ]; then
    echo "file2 already exists." else
    echo "file2 does not exist. Creating..."
    touch "dir1/file2"
    chmod a-rwx "dir1/file2"
    echo "file2 created." fi
```

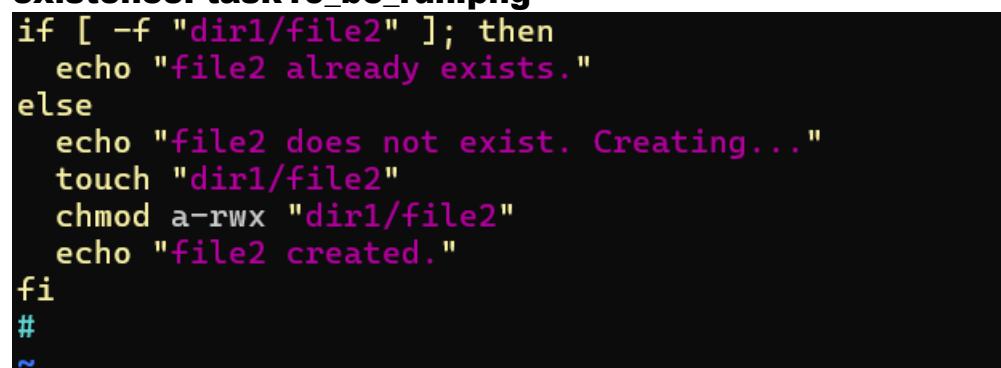
**Steps:**

vim setup.sh → append the code above → save and quit  
./setup.sh

**Screenshots:**

**vim editor showing file2 check code: task10\_b5\_vim.png**

**script run output showing file creation message or existence: task10\_b5\_run.png**



A terminal window titled "ubuntu@ubuntuserver: ~" showing the execution of a bash script. The script checks if a file file2 exists in the directory dir1. If it does not exist, it creates the file and sets its permissions. The output shows the file was created.

```
if [ -f "dir1/file2" ]; then
    echo "file2 already exists."
else
    echo "file2 does not exist. Creating..."
    touch "dir1/file2"
    chmod a-rwx "dir1/file2"
    echo "file2 created."
fi
#
~
```

```
ubuntu@ubuntuserver:~$ ./setup.sh
file2 does not exist. Creating...
file2 created.
ubuntu@ubuntuserver:~$ |
```

## 6. Check read, write, execute permissions on dir1/file2; grant missing user perms and show final ls

### Code to append:

```
# Permission checks for dir1/file2 (user permissions)
f="dir1/file2"if [ ! -r "$f" ]; then
    echo "Read permission missing; granting to user..."
    chmod u+r "$f"fi
if [ ! -w "$f" ]; then
    echo "Write permission missing; granting to user..."
    chmod u+w "$f"fi
if [ ! -x "$f" ]; then
    echo "Execute permission missing; granting to user..."
    chmod u+x "$f"fi
echo "Final permissions for $f:"
ls -l "$f"
```

### Steps:

vim setup.sh → append the code above → save and quit  
./setup.sh

### Screenshots:

**vim editor showing permission-check code: task10\_b6\_vim.png**  
**script run output showing the permission grants and final ls -l**  
**dir1/file2: task10\_b6\_run.png**

```
f="dir1/file2"
if [ ! -r "$f" ]; then
    echo "Read permission missing; granting to user..."
    chmod u+r "$f"fi
if [ ! -w "$f" ]; then
    echo "Write permission missing; granting to user..."
    chmod u+w "$f"fi
if [ ! -x "$f" ]; then
    echo "Execute permission missing; granting to user..."
    chmod u+x "$f"fi
echo "Final permissions for $f:"
ls -l "$f"
#
-- INSERT --
ubuntu@ubuntuserver:~$ ./setup.sh
file2 already exists.
Read permission missing; granting to user...
Write permission missing; granting to user...
Execute permission missing; granting to user...
Final permissions for dir1/file2:
-rwx----- 1 ubuntu ubuntu 0 Nov  3 11:04 dir1/file2
ubuntu@ubuntuserver:~$ |
```

## **Task 11 – Script setup.sh – argument comparisons (eq, ne, gt, lt, ge, le) and string checks**

### **1. create file with shebang and set num and str variables**

**In vim create/overwrite setup.sh and insert:**

```
#!/bin/bash  
num=$1  
str=$2
```

**Save and quit (:wq)**

**Make executable and run with examples:**

```
chmod +x setup.sh  
./setup.sh 10 Student
```

**Screenshots:**

**vim content: task11\_b0\_vim.png**

**run output: task11\_b0\_run.png**

A terminal window titled "ubuntu@ubuntuserver: ~" shows the creation and execution of a bash script named "setup.sh". The script defines variables num and str. It then contains a blank line, followed by a "#", a tilde (~), another tilde (~), and a final "#". The terminal then runs the command "chmod +x setup.sh" followed by "./setup.sh 10 Student". The output shows the script has run successfully.

```
ubuntu@ubuntuserver: ~  
#!/bin/bash  
|num=$1  
|str=$2  
  
#  
~  
~  
ubuntu@ubuntuserver:~$ chmod +x setup.sh  
. ./setup.sh 10 Student  
ubuntu@ubuntuserver:~$ |
```

### **2. add the -eq test (equal)**

**Append to setup.sh:**

```
if [ "$num" -eq 10 ]; then  
echo "$num is equal to 10 (-eq)."else  
echo "$num is NOT equal to 10 (-eq)."fi
```

**Save and quit; then run these commands (capture both in one terminal screenshot):**

```
./setup.sh 10 Student  
./setup.sh 7 Student
```

**Screenshots:**

**vim content after edit: task11\_b1\_vim.png**

**run output demonstrating both cases: task11\_b1\_run.png**

```
#!/bin/bash
num=$1
str=$2
if [ "$num" -eq 10 ]; then
    echo "$num is equal to 10 (-eq)."
else
    echo "$num is NOT equal to 10 (-eq)."
fi
#
~
~
```

ubuntu@ubuntuserver:~\$ ./setup.sh 10 Student  
10 is equal to 10 (-eq).  
ubuntu@ubuntuserver:~\$ ./setup.sh 7 Student  
7 is NOT equal to 10 (-eq).  
ubuntu@ubuntuserver:~\$ |

### 3. add the -ne test (not equal)

#### Append to setup.sh:

```
if [ "$num" -ne 10 ]; then
    echo "$num is not equal to 10 (-ne)."
else
    echo "$num is equal to 10 (-ne false)."
fi
```

#### Save and quit; run:

```
./setup.sh 7 Student
./setup.sh 10 Student
```

#### Screenshots:

**vim content: task11\_b2\_vim.png**

**run output: task11\_b2\_run.png**

```
if [ "$num" -ne 10 ]; then
    echo "$num is not equal to 10 (-ne)."
else
    echo "$num is equal to 10 (-ne false)."
fi
```

```
#
```

```
~
```

```
~
```

ubuntu@ubuntuserver:~\$ ./setup.sh 7 Student

7 is not equal to 10 (-ne).

ubuntu@ubuntuserver:~\$ ./setup.sh 10 Student

10 is equal to 10 (-ne false).

```
ubuntu@ubuntuserver:~$ |
```

### 4. add the -gt test (greater than)

#### Append:

```
if [ "$num" -gt 10 ]; then
```

```
echo "$num is greater than 10 (-gt)."else  
echo "$num is NOT greater than 10 (-gt)."fi
```

**Run:**

```
./setup.sh 12 Student  
./setup.sh 9 Student
```

**Screenshots:**

**vim content: task11\_b3\_vim.png**

**run output: task11\_b3\_run.png**

```
if [ "$num" -gt 10 ]; then  
  echo "$num is greater than 10 (-gt)."br/>else  
  echo "$num is NOT greater than 10 (-gt)."br/>fi  
|  
#  
~
```

```
ubuntu@ubuntuserver:~$ ./setup.sh 12 Student  
.setup.sh 9 Student  
12 is greater than 10 (-gt).  
9 is NOT greater than 10 (-gt).  
ubuntu@ubuntuserver:~$ |
```

**5. add the -lt test (less than)**

**Append:**

```
if [ "$num" -lt 10 ]; then  
  echo "$num is less than 10 (-lt)."else  
  echo "$num is NOT less than 10 (-lt)."fi
```

**Run:**

```
./setup.sh 5 Student  
./setup.sh 11 Student
```

**Screenshots:**

**vim content: task11\_b4\_vim.png**

**run output: task11\_b4\_run.png**

```
if [ "$num" -lt 10 ]; then  
  echo "$num is less than 10 (-lt)."br/>else  
  echo "$num is NOT less than 10 (-lt)."br/>fi  
|  
#
```

```
ubuntu@ubuntuserver:~$ ./setup.sh 5 Student  
.setup.sh 11 Student  
5 is less than 10 (-lt).  
11 is NOT less than 10 (-lt).  
ubuntu@ubuntuserver:~$ |
```

**6. add the -ge test (greater than or equal)**

**Append:**

```
if [ "$num" -ge 10 ]; then
```

```
echo "$num is greater than or equal to 10 (-ge)."else  
echo "$num is NOT greater than or equal to 10 (-ge)."fi
```

**Run:**

```
./setup.sh 10 Student  
./setup.sh 8 Student
```

**Screenshots:**

**vim content: task11\_b5\_vim.png**

**run output: task11\_b5\_run.png**

```
if [ "$num" -ge 10 ]; then  
    echo "$num is greater than or equal to 10 (-ge)."else  
    echo "$num is NOT greater than or equal to 10 (-ge)."fi  
|
```

```
ubuntu@ubuntuserver:~$ ./setup.sh 10 Student  
.setup.sh 8 Student  
10 is greater than or equal to 10 (-ge).  
8 is NOT greater than or equal to 10 (-ge).  
ubuntu@ubuntuserver:~$ |
```

**7. add the -le test (less than or equal)**

**Append:**

```
if [ "$num" -le 10 ]; then  
    echo "$num is less than or equal to 10 (-le)."else  
    echo "$num is NOT less than or equal to 10 (-le)."fi
```

**Run:**

```
./setup.sh 10 Student  
./setup.sh 12 Student
```

**Screenshots:**

**vim content: task11\_b6\_vim.png**

**run output: task11\_b6\_run.png**

```
if [ "$num" -le 10 ]; then  
    echo "$num is less than or equal to 10 (-le)."else  
    echo "$num is NOT less than or equal to 10 (-le)."fi  
|
```

```
ubuntu@ubuntuserver:~$ ./setup.sh 10 Student  
.setup.sh 12 Student  
10 is less than or equal to 10 (-le).  
12 is NOT less than or equal to 10 (-le).  
ubuntu@ubuntuserver:~$ |
```

**8. string equality test ( = )**

**Ensure str=\$2 exists at top (1.).**

**Append:**

```
if [ "$str" = "Student" ]; then  
    echo "Second argument equals 'Student' ( = )."else
```

```
echo "Second argument does NOT equal 'Student' ( = )."fi
```

**Run:**

```
./setup.sh 10 Student  
./setup.sh 10 Test
```

**Screenshots:**

**vim content: task11\_b7\_vim.png**

**run output: task11\_b7\_run.png**

```
if [ "$str" = "Student" ]; then  
    echo "Second argument equals 'Student' ( = )."  
else  
    echo "Second argument does NOT equal 'Student' ( = )."  
fi  
|
```

```
ubuntu@ubuntuserver:~$ ./setup.sh 10 Student  
./setup.sh 10 Test  
Second argument equals 'Student' ( = ).  
Second argument does NOT equal 'Student' ( = ).  
ubuntu@ubuntuserver:~$ |
```

**9. string inequality test ( != )**

**Append:**

```
if [ "$Str" != "Student" ]; then  
    echo "Second argument is not equal to 'Student' ( != )."  
else  
    echo "Second argument equals 'Student' ( != false)."  
fi
```

**Run:**

```
./setup.sh 10 Test  
./setup.sh 10 Student
```

**Screenshots:**

**vim content: task11\_b8\_vim.png**

**run output: task11\_b8\_run.png**

```
if [ "$Str" != "Student" ]; then  
    echo "Second argument is not equal to 'Student' ( != )."  
else  
    echo "Second argument equals 'Student' ( != false)."  
fi  
|  
#  
~  
~
```

```
ubuntu@ubuntuserver:~$ ./setup.sh 10 Test  
./setup.sh 10 Student  
Second argument is not equal to 'Student' ( != ).  
Second argument equals 'Student' ( != false).  
ubuntu@ubuntuserver:~$ |
```

**10. check if second argument is empty (zero-length)**

**Append:**

```
if [ -z "$str" ]; then  
    echo "Second argument is empty (zero-length)."  
else  
    echo "Second argument is not empty."  
fi
```

**Run:**

```
./setup.sh 10  
./setup.sh 10 Student
```

**Screenshots:****vim content: task11\_b9\_vim.png****run output: task11\_b9\_run.png**

```
if [ -z "$str" ]; then  
    echo "Second argument is empty (zero-length)."  
else  
    echo "Second argument is not empty."  
fi  
  
#  
ubuntu@ubuntuserver:~$ ./setup.sh 10  
./setup.sh 10 Student  
Second argument is empty (zero-length).  
Second argument is not empty.  
ubuntu@ubuntuserver:~$ |
```

**Task 12 – Script setup.sh – print all arguments with a for loop****1. Create the script with shebang and basic structure****Open vim and overwrite setup.sh:****vim setup.sh****Insert these lines (first step — shebang and a short comment):**

```
#!/bin/bash# Script to demonstrate printing all user-entered arguments using  
$*
```

Save and quit (:wq)

**Screenshots:****vim editor showing the shebang and****comment: task12\_b1\_vim.png****run (no output expected but show ./setup.sh****run): task12\_b1\_run.png**

```
ubuntu@ubuntuserver: ~ x + v  
#!/bin/bash  
# Script to demonstrate printing all user-entered arguments using $*  
#  
~  
ubuntu@ubuntuserver:~$ chmod +x setup.sh  
./setup.sh  
ubuntu@ubuntuserver:~$ |
```

**2. Append the for loop using \$\* and print each argument****Re-open setup.sh in vim and append the following lines:**

```
# Print all arguments using $*echo "Printing all arguments using \$*:."for arg in  
$*; do
```

```
    echo "Argument: $arg"done
```

Save and quit (:wq)

**Make the script executable and run it with example arguments:**

```
chmod +x setup.sh  
./setup.sh one "two words" three
```

**Screenshots:**

**vim editor showing the for-loop appended: task12\_b2\_vim.png**

**script run output showing the printed**

**arguments: task12\_b2\_run.png**

```
# Print all arguments using $*  
echo "Printing all arguments using \$*:  
for arg in $*; do  
    echo "Argument: $arg"  
done  
#  
ubuntu@ubuntuserver:~$ ./setup.sh one "two words" three  
Printing all arguments using $*:  
Argument: one  
Argument: two  
Argument: words  
Argument: three  
ubuntu@ubuntuserver:~$ |
```

**Task 13 – Script setup.sh – while loop summation and functions**

**1. Add the shebang line**

**Open vim and overwrite setup.sh with the shebang line:**

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

**Save and quit (:wq)**

**Make executable and run (no output expected):**

```
chmod +x setup.sh  
./setup.sh
```

**Screenshots:**

**vim editor showing shebang: task13\_b1\_vim.png**

**run output: task13\_b1\_run.png**

```
#!/bin/bash  
#  
~  
~  
~  
~  
ubuntu@ubuntuserver:~$ chmod +x setup.sh  
./setup.sh  
ubuntu@ubuntuserver:~$ |
```

## **2. Add the while-loop summation (interactive)**

**Re-open setup.sh in vim and append the while-loop:**

```
# While-loop summation (interactive)
sum=0while true; do
    read -p "Enter a number (or 'q' to quit): " input
    if [ "$input" = "q" ]; then
        break
    fi

    sum=$((sum + input))
    echo "Total Score: $sum"doneecho "Final total: $sum"
```

**Save and quit (:wq)**

**Run the script and demonstrate a short session (example):**

**enter 5, then 7, then q**

```
./setup.sh# interactively enter:# 5# 7# q
```

**Screenshots:**

**vim editor showing while-loop appended: task13\_b2\_vim.png**

**run output showing the interactive session and**

**totals: task13\_b2\_run.png**

```
# While-loop summation (interactive)
sum=0
while true; do
    read -p "Enter a number (or 'q' to quit): " input
    if [ "$input" = "q" ]; then
        break
    fi

    sum=$((sum + input))
    echo "Total Score: $sum"
done
echo "Final total: $sum"
|
```

```
#ubuntu@ubuntuserver:~$ ./setup.sh
Enter a number (or 'q' to quit): 5
Total Score: 5
Enter a number (or 'q' to quit): 3
Total Score: 8
Enter a number (or 'q' to quit): q
Final total: 8
ubuntu@ubuntuserver:~$ |
```

## **3. Add the interactive summation function and demonstrate it**

**Re-open setup.sh in vim and append the  
function sum\_two() which contains its own interactive while-  
loop:**

```
# Function to accumulate scores interactivelysum_two() {
    sum=0
    while true; do
```

```

read -p "Enter a number (or 'q' to quit): " input
if [ "$input" = "q" ]; then
    break
fi

sum=$((sum + input))
echo "Total Score: $sum"
done
echo "Function final total: $sum"
}

```

**# Demonstrate the function echo "Now calling sum\_two function:"**

sum\_two

**Save and quit (:wq)**

**Important: If you have the standalone while-loop from step 2 and you place this function into the script, delete the standalone loop to avoid executing the same interactive logic twice when running the script.**

**Run the script and demonstrate a short session (example):**

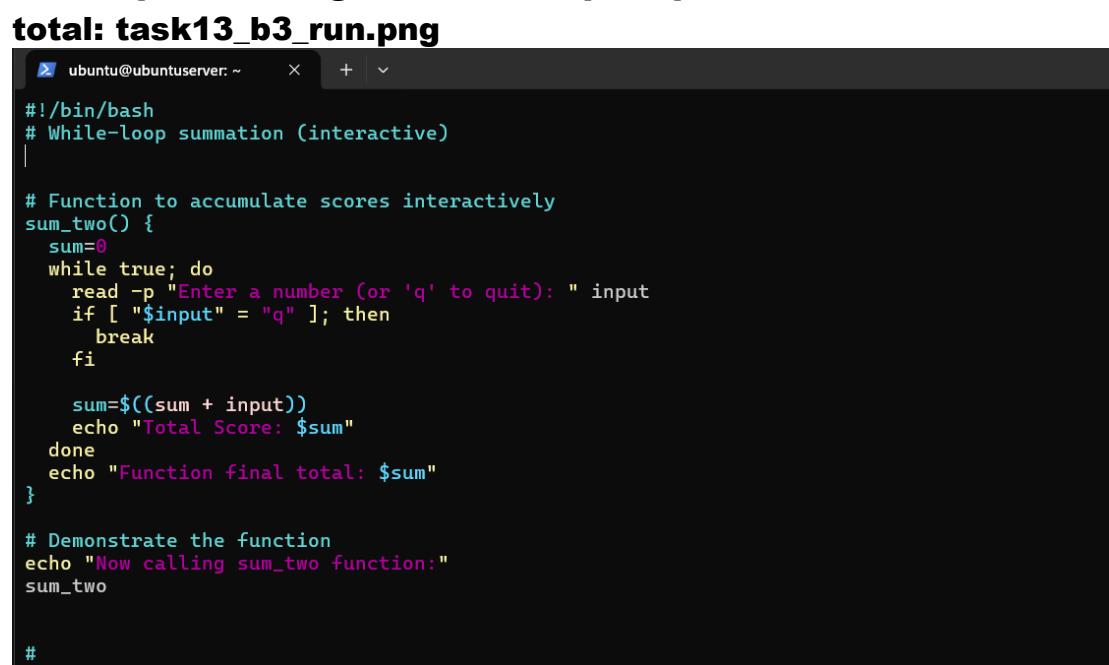
**enter 3, 4, q when prompted by the function:**

./setup.sh# when prompted by the function enter:# 3# 4# q

**Screenshots:**

**vim editor showing function appended: task13\_b3\_vim.png**

**run output showing the function prompts and final total: task13\_b3\_run.png**



```

ubuntu@ubuntuserver: ~      + | ^
#!/bin/bash
# While-loop summation (interactive)

# Function to accumulate scores interactively
sum_two() {
    sum=0
    while true; do
        read -p "Enter a number (or 'q' to quit): " input
        if [ "$input" = "q" ]; then
            break
        fi

        sum=$((sum + input))
        echo "Total Score: $sum"
    done
    echo "Function final total: $sum"
}

# Demonstrate the function
echo "Now calling sum_two function:"
sum_two

#

```

```
ubuntu@ubuntuserver:~$ ./setup.sh
Now calling sum_two function:
Enter a number (or 'q' to quit): 4
Total Score: 4
Enter a number (or 'q' to quit): 2
Total Score: 6
Enter a number (or 'q' to quit): q
Function final total: 6
ubuntu@ubuntuserver:~$ |
```

**4. Add a function that takes two numeric arguments, sums them, and returns the result (echo)**

**Re-open setup.sh in vim and append the following function and demonstration. This function accepts two numeric arguments, adds them, and return the sum. The script then captures that output and displays it.**

**# Function that sums two arguments and returns the result**

```
sum_args() {
    a=$1
    b=$2
    return $((a + b))
}
```

**# Demonstrate sum\_args function**  
**echo "Now demonstrating sum\_args function:"**

```
sum_args 3 4
result=$?echo "sum_args(3,4) returned: $result"
```

**Save and quit (:wq)**

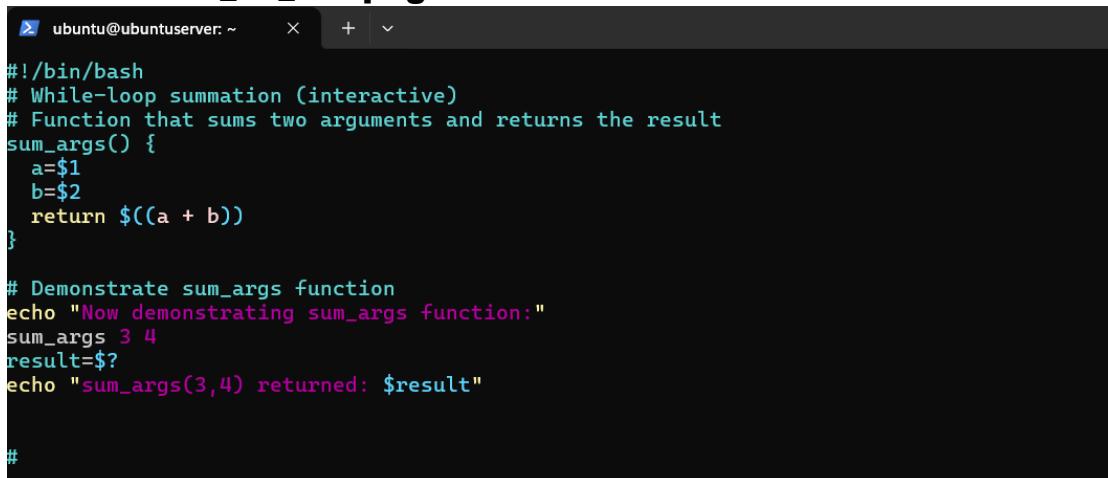
**Run the script and capture the demonstration output:**

```
chmod +x setup.sh
```

```
./setup.sh# Observe the output that shows "sum_args(3,4) returned: 7"
```

**Screenshots:**

**vim editor showing function appended: task13\_b4\_vim.png**  
**run output showing function demonstration and returned sum: task13\_b4\_run.png**



```
ubuntu@ubuntuserver: ~      x  +  v
#!/bin/bash
# While-Loop summation (interactive)
# Function that sums two arguments and returns the result
sum_args() {
    a=$1
    b=$2
    return $((a + b))
}

# Demonstrate sum_args function
echo "Now demonstrating sum_args function:"
sum_args 3 4
result=$?
echo "sum_args(3,4) returned: $result"

#
```

```
ubuntu@ubuntuserver:~$ ./setup.sh
Now demonstrating sum_args function:
sum_args(3,4) returned: 7
```

## Task 14 – Codespaces GUI — fork repo, run start-desktop.sh, open VNC, stop GUI

### Steps:

#### 1. Fork the repository to your GitHub account

Open the repo URL in your browser:

**Ubuntu Machine**

**Click "Fork" (top-right) and fork it to your account.**

**Save screenshot as: task14\_fork.png**

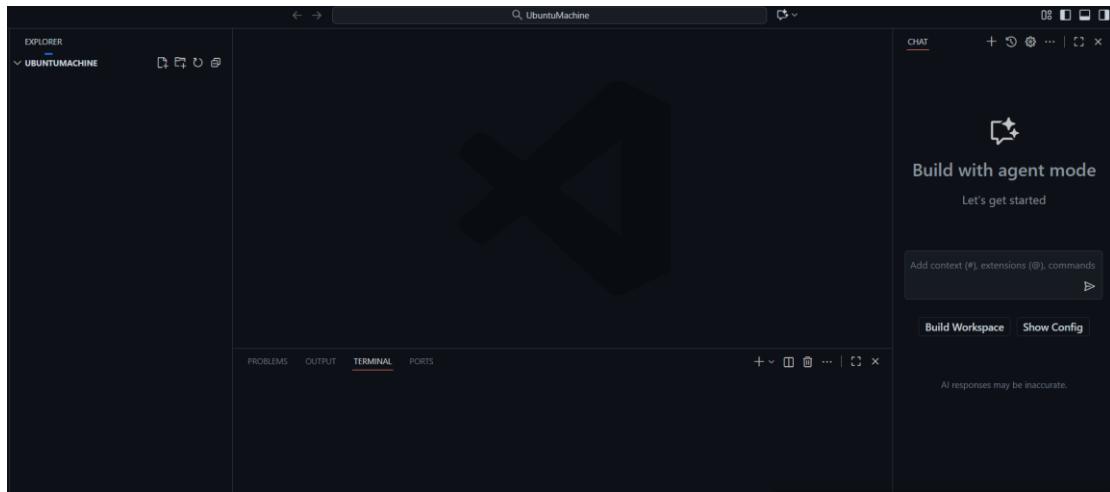
The screenshot shows the 'Create a new fork' page on GitHub. The 'Owner' field is set to 'lujainzia127'. The 'Repository name' field contains 'UbuntuMachine' with a note below it stating 'UbuntuMachine is available.' A description field is present but empty. A checked checkbox 'Copy the main branch only' has a note below it: 'Contribute back to WaqasSaleem97/UbuntuMachine by adding your own branch.' A note at the bottom left says '(i) You are creating a fork in your personal account.' A green 'Create fork' button is at the bottom right.

#### 2. Open a Codespace on your fork

**In your forked repository on GitHub, click the green "Code" button → "Open with Codespaces" → "Create codespace on main" (or appropriate branch).**

**Wait for the Codespace to initialize.**

**Save screenshot as: task14\_codespace\_launch.png**



### **3. Verify the start script is present and executable (capture evidence)**

**In the Codespace terminal list files in the repo root and show the start script and stop script exist:**

ls -l start-desktop.sh stop-desktop.sh

**If not executable, make it executable:**

chmod +x start-desktop.sh stop-desktop.sh

**Save a screenshot showing the ls -l output (file listing) and the chmod command if applied:**

**task14\_start\_script\_ls.png**

```
@lujainzia127 →/workspaces/UbuntuMachine (main) $ ls -l start-desktop.sh stop-desktop.sh
-rwxrwxrwx 1 codespace root 1333 Nov  3 12:47 start-desktop.sh
-rwxrwxrwx 1 codespace root  428 Nov  3 12:47 stop-desktop.sh
@lujainzia127 →/workspaces/UbuntuMachine (main) $
```

### **4. Run the start script inside the Codespace terminal**

**In the Codespace terminal run:**

**# Ensure the start script is executable**

chmod +x start-desktop.sh

**# Start the desktop GUI**

./start-desktop.sh

**Capture the terminal output showing successful start messages.**

**Save screenshot as: task14\_start\_run.png**

```
@lujainzia127 →/workspaces/UbuntuMachine (main) $ WebSocket server settings:
- Listen on :6080
- Web server. Web root: /usr/share/novnc
- No SSL/TLS support (no cert file)
- proxying from :6080 to localhost:5901
```

### **5. Verify forwarded ports in Codespaces (Ports view)**

**Open the Codespaces "Ports" panel / view and confirm port 6080 is forwarded and visible.**

**Save a screenshot of the Ports view showing port 6080 and its status:**

**task14\_ports\_view.png**

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS 3
Port	Forwarded Addr...	Running Process	Visibility	Origin
● 5900	https://ominous-o...	x11vnc -display :1 -rfbauth /home/codes...	🔒 Private	Auto Forwarded
● 5901	https://ominous-o...	x11vnc -display :1 -rfbauth /home/codes...	🔒 Private	Auto Forwarded
● 6080	https://ominous-o...	/usr/bin/python3 /usr/bin/websockify ---	🔒 Private	Auto Forwarded

[Add Port](#)

## 6. Open forwarded port 6080 and connect to VNC HTML page

In the Codespaces UI, open the forwarded port's preview URL or copy the forwarded URL and open it in your browser.

Visit the port 6080 address and click the vnc.html link.

When prompted for a password enter:

codespace

Capture screenshots of:

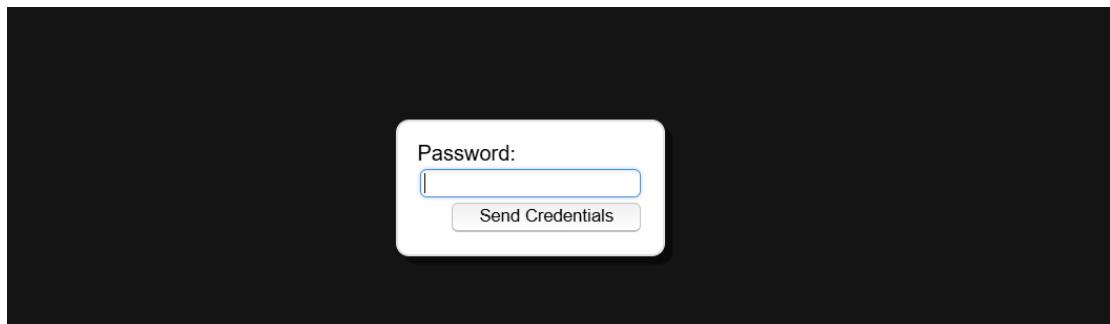
The browser showing the forwarded port URL in the address bar / Codespaces preview: task14\_vnc\_url.png

## Directory listing for /

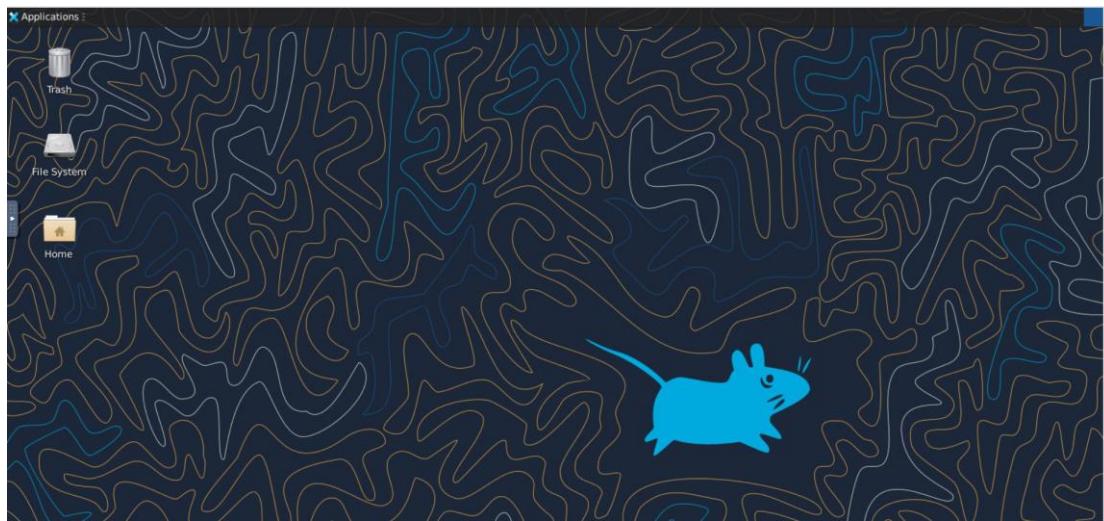
---

- [app/](#)
  - [core/](#)
  - [include/](#)
  - [utils/](#)
  - [vendor/](#)
  - [vnc.html](#)
  - [vnc\\_auto.html@](#)
  - [vnc\\_lite.html](#)
- 

The VNC password prompt (showing password field; do NOT include typed password in a screenshot): task14\_vnc\_password\_prompt.png



The VNC session after successful connection showing the GUI/desktop: task14\_vnc\_desktop.png



**(Optional) A focused screenshot of vnc.html UI showing the "Connect" button before/after connecting: task14\_vnc\_connect.png**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 3 bash + ⌂ ⌂ ... | ⌂ x
03/11/2025 13:02:31 incr accepted_client=2 for 127.0.0.1:48480 sock=13
03/11/2025 13:02:32 Client Protocol Version 3.8
03/11/2025 13:02:32 Protocol version sent 3.8, using 3.8
03/11/2025 13:02:32 rfbProcessClientSecurityType: executing handler for type 2
03/11/2025 13:03:12 client_count: 1
03/11/2025 13:03:12 Client 127.0.0.1 gone
03/11/2025 13:03:12 Statistics          events      Transmit/ RawEquiv ( saved)
03/11/2025 13:03:12 TOTALS            :      0 |        0/        0 (  0.0%)
03/11/2025 13:03:12 Statistics          events      Received/ RawEquiv ( saved)
03/11/2025 13:03:12 TOTALS            :      0 |        0/        0 (  0.0%)
```

## 7. Stop the GUI

**When finished, return to the Codespace terminal and run:  
./stop-desktop.sh**

**Capture the terminal output that shows the GUI stopping and any cleanup messages.**

**Save screenshot as: task14\_stop\_run.png**

```
@lujainzia127 → /workspaces/UbuntuMachine (main) $ WebSocket server settings:  
- Listen on :6080  
- Web server. Web root: /usr/share/novnc  
- No SSL/TLS support (no cert file) ...  
  
** (xfce4-power-manager:22016): WARNING **: 13:04:58.136: Error: The connection is closed  
  
** (xfce4-power-manager:22016): WARNING **: 13:04:58.136: Error: The connection is closed  
  
xfsettingsd: Another instance took over. Leaving...  
@lujainzia127 → /workspaces/UbuntuMachine (main) $
```

## **Exam evaluation question:**

## **Exam Evaluation Questions**

**Use the format below for each exam evaluation question. Each question includes a short scenario and clear numbered steps students must perform; capture the requested screenshots as**

**evidence. These questions cover the concepts taught throughout this lab (users/groups, account files, ownership/permissions, pipes/grep/redirects, scripting basics, conditionals/comparisons, loops/functions, Codespaces/VNC). Do NOT include answers or solutions in this file.**

---

## 1. Group Management and Membership

### Scenario:

**Create groups and manage a user's primary and supplementary group memberships.**

### Steps:

**Create groups g1, g2, and g3.**

**Screenshot: Q1\_groups\_created.png**

```
ubuntu@ubuntuserver:~$ sudo groupadd g1
sudo groupadd g2
sudo groupadd g3
[sudo] password for ubuntu:
ubuntu@ubuntuserver:~$ |
```

**Change examuser's primary group to g3 and add g1 and g2 as supplementary groups.**

**Screenshot: Q1\_group\_changes.png**

```
ubuntu@ubuntuserver:~$ sudo usermod -g g3 examuser
ubuntu@ubuntuserver:~$ sudo usermod -aG g1,g2 examuser
ubuntu@ubuntuserver:~$ |
```

**Show the final id and /etc/group lines that prove the changes.**

**Screenshot: Q1\_group\_verification.png**

```
ubuntu@ubuntuserver:~$ id examuser
grep 'g[123]' /etc/group
uid=1009(examuser) gid=1008(g3) groups=1008(g3),100(users),1006(g1),1007(g2)
g1:x:1006:examuser
g2:x:1007:examuser
g3:x:1008:
ubuntu@ubuntuserver:~$ |
```

---

## 2. Ownership and Permission Tasks

### Scenario:

**Demonstrate ownership changes and apply both symbolic and numeric permission changes.**

### Steps:

**Create workspace/secret.txt, change its owner to examuser and group to g1.**

**Screenshot: Q2\_chown\_chgrp.png**

```
ubuntu@ubuntuserver:~$ ls -l ~/workspace/secret.txt
-rw-rw-r-- 1 examuser g1 0 Nov  3 13:12 /home/ubuntu/workspace/secret.txt
ubuntu@ubuntuserver:~$ |
```

**Remove all permissions for group and others using a symbolic command, then using a numeric command to achieve the same result.**

**Screenshot: Q2\_symbolic\_numeric.png**

```
ubuntu@ubuntuserver:~$ sudo chmod 700 ~/workspace/secret.txt
ubuntu@ubuntuserver:~$ sudo chmod go-rwx ~/workspace/secret.txt
ubuntu@ubuntuserver:~$ ls -l ~/workspace/secret.txt
-rwx----- 1 examuser g1 0 Nov  3 13:12 /home/ubuntu/workspace/secret.txt
ubuntu@ubuntuserver:~$ |
```

**Show ls -l for the file after each change to document the permission bits.**

**Screenshot: Q2\_permissions\_ls.png**

```
ubuntu@ubuntuserver:~$ ls -l ~/workspace/secret.txt
-rwx----- 1 examuser g1 0 Nov  3 13:12 /home/ubuntu/workspace/secret.txt
ubuntu@ubuntuserver:~$ |
```

---

### 3. Pipes, Grep, and Redirection Practice

**Scenario:**

**Filter system logs and save results using redirection and piping.**

**Steps:**

**Use grep (or journalctl where applicable) with a pipe to find lines containing "error" or "fail" and show the first 20 results.**

**Screenshot: Q3\_grep\_pipe.png**

```
ubuntu@ubuntuserver:~$ grep -Ei 'error|fail' /var/log/syslog | head -n 20
2025-11-01T12:00:45.108493+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.107789122Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.aufs\"..." error="aufs is not supported (modprobe aufs failed: exit sta
tus 1 \\"modprobe: FATAL: Module aufs not found in directory /lib/modules/6.8.0-86-generic\\n\\"): skip plugin" type=io.co
ntainerd.snapshotter.v1
2025-11-01T12:00:45.114223+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113006059Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.blockfile\"..." error="no scratch file generator: skip plugin" type=io.c
ontainerd.snapshotter.v1
2025-11-01T12:00:45.114521+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113611499Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.btrfs\"..." error="path /var/lib/containerd/io.containerd.snapshotter.v
1.btrfs (ext4) must be a btrfs filesystem to be used with the btrfs snapshotter: skip plugin" type=io.containerd.snapsho
tter.v1
2025-11-01T12:00:45.114620+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113667999Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.devmapper\"..." error="devmapper not configured: skip plugin" type=io.c
ontainerd.snapshotter.v1
2025-11-01T12:00:45.117241+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.116617183Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.zfs\"..." error="path /var/lib/containerd/io.containerd.snapshotter.v1.
zfs must be a zfs filesystem to be used with the zfs snapshotter: skip plugin" type=io.containerd.snapshotter.v1
2025-11-01T12:00:45.157755+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.152569411Z" level=info msg="sk
ip loading plugin \"io.containerd.tracing.processor.v1.otlp\"..." error="skip plugin: tracing endpoint not configured" t
ype=io.containerd.tracing.processor.v1
2025-11-01T12:00:45.158436+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.152712262Z" level=info msg="sk
ip loading plugin \"io.containerd.internal.v1.tracing\"..." error="skip plugin: tracing endpoint not configured" type=io.
containerd.internal.v1
2025-11-01T12:00:47.359789+00:00 ubuntuserver dockerd[1666]: time="2025-11-01T12:00:47.358707171Z" level=info msg="CDI d
irectory does not exist, skipping: failed to monitor for changes: no such file or directory" dirs=/etc/cdi
2025-11-01T12:00:47.359883+00:00 ubuntuserver dockerd[1666]: time="2025-11-01T12:00:47.358787084Z" level=info msg="CDI d
irectory does not exist, skipping: failed to monitor for changes: no such file or directory" dirs=/var/run/cdi
2025-11-01T12:00:48.272164+00:00 ubuntuserver multipath: sda: failed to get sysfs uid: No such file or directory
```

**Save the filtered results to a file ~logs/errors.txt using overwrite, then append additional matching lines using append redirection.**

**Screenshot: Q3\_redirect\_overwrite\_append.png**

```
ubuntu@ubuntuserver:~$ mkdir -p ~/logs
grep -Ei 'error|fail' /var/log/syslog > ~/logs/errors.txt
ubuntu@ubuntuserver:~$ grep -Ei 'warn|critical' /var/log/syslog >> ~/logs/errors.txt
ubuntu@ubuntuserver:~$
```

## Use a pager to view the saved file.

**Screenshot: Q3\_pager\_view.png**

```
ubuntu@ubuntuserver:~  x + v
2025-11-01T12:00:45.108493+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.107789122Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.aufs\"..." error="aufs is not supported (modprobe aufs failed: exit sta
tus 1 \\"modprobe: FATAL: Module aufs not found in directory /lib/modules/6.8.0-86-generic\\n\\"): skip plugin" type=io.co
ntainerd.snapshotter.v1
2025-11-01T12:00:45.114223+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113006059Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.blockfile\"..." error="no scratch file generator: skip plugin" type=io.
containerd.snapshotter.v1
2025-11-01T12:00:45.117241+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113611499Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.btrfs\"..." error="path /var/lib/containerd/io.containerd.snapshotter.v
1.btrfs (ext4) must be a btrfs filesystem to be used with the btrfs snapshotter: skip plugin" type=io.containerd.snapshot
tter.v1
2025-11-01T12:00:45.114620+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.113667991Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.devmapper\"..." error="devmapper not configured: skip plugin" type=io.c
ontainerd.snapshotter.v1
2025-11-01T12:00:45.117241+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.116617183Z" level=info msg="sk
ip loading plugin \"io.containerd.snapshotter.v1.zfs\"..." error="path /var/lib/containerd/io.containerd.snapshotter.v1.
zfs must be a zfs filesystem to be used with the zfs snapshotter: skip plugin" type=io.containerd.snapshotter.v1
2025-11-01T12:00:45.157755+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.152569411Z" level=info msg="sk
ip loading plugin \"io.containerd.tracing.processor.v1.otp\"..." error="skip plugin: tracing endpoint not configured" t
ype=io.containerd.tracing.processor.v1
2025-11-01T12:00:45.158436+00:00 ubuntuserver containerd[1457]: time="2025-11-01T12:00:45.152712262Z" level=info msg="sk
ip loading plugin \"io.containerd.internal.v1.tracing\"..." error="skip plugin: tracing endpoint not configured" type=io.
containerd.internal.v1
2025-11-01T12:00:47.359789+00:00 ubuntuserver dockerd[1666]: time="2025-11-01T12:00:47.358707171Z" level=info msg="CDI d
irectory does not exist, skipping: failed to monitor for changes: no such file or directory" dirs=/etc/cdi
2025-11-01T12:00:47.359883+00:00 ubuntuserver dockerd[1666]: time="2025-11-01T12:00:47.358787084Z" level=info msg="CDI d
irectory does not exist, skipping: failed to monitor for changes: no such file or directory" dirs=/var/run/cdi
2025-11-01T12:00:48.272164+00:00 ubuntuserver multipath: sda: failed to get sysfs uid: No such file or directory
2025-11-01T12:00:48.272357+00:00 ubuntuserver multipath: sda: failed to get sgio uid: No such file or directory
/home/ubuntu/logs/errors.txt
```

---

## 4. Script: Variables, Command Substitution, File & Dir Checks

### Scenario:

**Build and run a script incrementally that demonstrates variables, command substitution, and filesystem checks.**

### Steps:

**Create setup.sh with a shebang and a variable var1 that you echo.**

**Screenshot: Q4\_step1\_var1.png**

```
ubuntu@ubuntuserver:~$ vim setup.sh
ubuntu@ubuntuserver:~$ chmod +x setup.sh
./setup.sh
Hello from var1
ubuntu@ubuntuserver:~$ |
```

**Append command substitution that stores ls -l output into a variable and echo it.**

**Screenshot: Q4\_step2\_allfiles.png**

```

ubuntu@ubuntuserver:~$ ./setup.sh
All files listed:
total 411704
drwxrwxr-x 3 ubuntu ubuntu      4096 Oct 20 13:22 analysis_lab
-rw-rw-r-- 1 ubuntu ubuntu      482 Oct 18 19:36 answers.md
-rw-rw-r-- 1 ubuntu ubuntu      417 Oct 24 08:47 apt_update_vs_upgrade.md
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 25 12:31 Desktop
drwxrwxr-x 2 ubuntu ubuntu      4096 Nov  3 11:04 dir1
-rw-rw-r-- 1 ubuntu ubuntu 421490404 Oct 23 14:32 docker-desktop-amd64.deb
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Documents
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Downloads
drwxrwxr-x 3 ubuntu ubuntu      4096 Oct 20 13:26 evidence_backup
drwxrwxr-x 3 ubuntu ubuntu      4096 Oct 18 19:44 lab4
drwxrwxr-x 2 ubuntu ubuntu      4096 Oct 24 20:48 Lab5
drwxrwxr-x 2 ubuntu ubuntu      4096 Oct 24 20:09 Lab5cd
drwxrwxr-x 2 ubuntu ubuntu      4096 Nov  3 13:27 logs
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Music
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Pictures
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Public
-rw-rw-r-- 1 ubuntu ubuntu      179 Oct 20 13:22 Q2_report.md
-rwxrwxr-x 1 ubuntu ubuntu      73 Nov  3 13:40 setup.sh
drwx----- 3 ubuntu ubuntu      4096 Oct 24 09:01 snap
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Templates
drwxrwxr-t 2 ubuntu ubuntu      4096 Oct 24 09:15 thinclient_drives
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Videos
drwxrwxr-x 2 ubuntu ubuntu      4096 Nov  3 13:12 workspace
ubuntu@ubuntuserver:~$ |

```

**Append directory and file checks that  
create dir1 and dir1/file2 if missing, and display their final  
permissions.**

#### Screenshot: Q4\_step3\_dirfile\_checks.png

```

All files listed:
total 411704
drwxrwxr-x 3 ubuntu ubuntu      4096 Oct 20 13:22 analysis_lab
-rw-rw-r-- 1 ubuntu ubuntu      482 Oct 18 19:36 answers.md
-rw-rw-r-- 1 ubuntu ubuntu      417 Oct 24 08:47 apt_update_vs_upgrade.md
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 25 12:31 Desktop
drwxrwxr-x 2 ubuntu ubuntu      4096 Nov  3 11:04 dir1
-rw-rw-r-- 1 ubuntu ubuntu 421490404 Oct 23 14:32 docker-desktop-amd64.deb
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Documents
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Downloads
drwxrwxr-x 3 ubuntu ubuntu      4096 Oct 20 13:26 evidence_backup
drwxrwxr-x 3 ubuntu ubuntu      4096 Oct 18 19:44 lab4
drwxrwxr-x 2 ubuntu ubuntu      4096 Oct 24 20:48 Lab5
drwxrwxr-x 2 ubuntu ubuntu      4096 Oct 24 20:09 Lab5cd
drwxrwxr-x 2 ubuntu ubuntu      4096 Nov  3 13:27 logs
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Music
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Pictures
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Public
-rw-rw-r-- 1 ubuntu ubuntu      179 Oct 20 13:22 Q2_report.md
-rwxrwxr-x 1 ubuntu ubuntu      288 Nov  3 13:42 setup.sh
drwx----- 3 ubuntu ubuntu      4096 Oct 24 09:01 snap
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Templates
drwxrwxr-t 2 ubuntu ubuntu      4096 Oct 24 09:15 thinclient_drives
drwxr-xr-x 2 ubuntu ubuntu      4096 Oct 24 09:15 Videos
drwxrwxr-x 2 ubuntu ubuntu      4096 Nov  3 13:12 workspace
Permissions of dir1:
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov  3 11:04 dir1
Permissions of file2:
-rwx----- 1 ubuntu ubuntu 0 Nov  3 11:04 dir1/file2
ubuntu@ubuntuserver:~$ |

```

## 5. Script: Comparisons and String Tests

### Scenario:

**Incrementally add numeric and string comparison tests to a script and show both true/false cases.**

### Steps:

**Overwrite setup.sh to set num=\$1 and str=\$2, and add an -eq test showing true and false examples.**

**Screenshot: Q5\_eq\_examples.png**

```
ubuntu@ubuntuserver:~$ ./setup.sh 5 test
./setup.sh 3 test
Number equals 5
Number is not 5
ubuntu@ubuntuserver:~$ |
```

**Append -ne, -gt, -lt, -ge, and -le tests and demonstrate at least one true and one false invocation for each.**

**Screenshot: Q5\_numeric\_tests.png**

```
ubuntu@ubuntuserver:~$ vim setup.sh
ubuntu@ubuntuserver:~$ ./setup.sh 6 test
./setup.sh 2 test
Number is not 5
Not 4
Greater than 3
Less than 10
>=5
<=7
Number is not 5
Not 4
Not greater
Less than 10
<5
<=7
ubuntu@ubuntuserver:~$ |
```

**Append string equality (=) and inequality (!=) checks and a -z (zero-length) test for the second argument, demonstrating true/false cases.**

**Screenshot: Q5\_string\_tests.png**

```
ubuntu@ubuntuserver:~$ ./setup.sh ""
Not hello
Not world
Empty string
ubuntu@ubuntuserver:~$ ./setup.sh hello
String is hello
Not world
Not empty
ubuntu@ubuntuserver:~$ |
```

---

## 6. Script: For Loop and Argument Handling

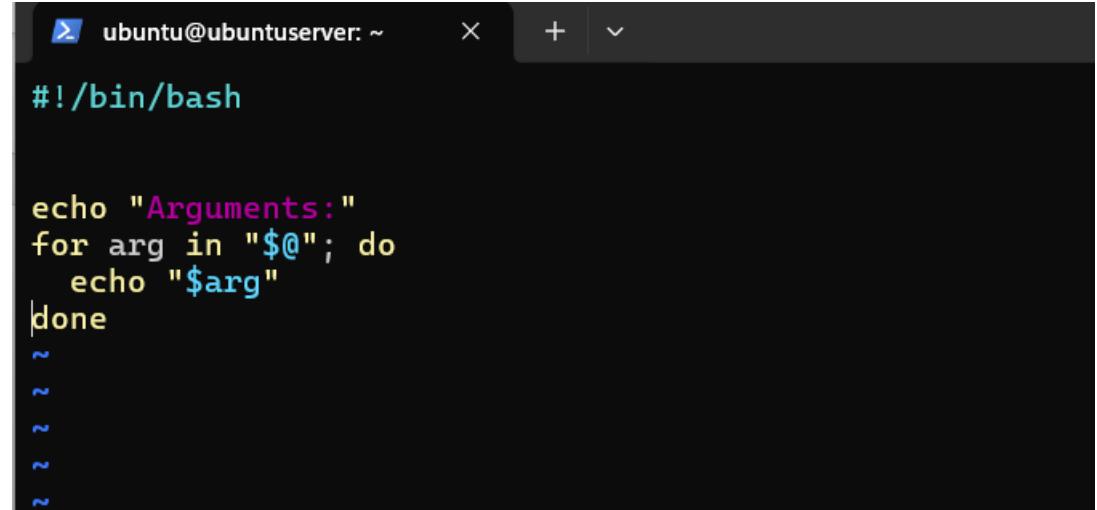
**Scenario:**

**Write a script that prints all provided arguments and demonstrate correct handling of quoted multi-word arguments.**

**Steps:**

**Create/overwrite setup.sh to print every argument using "\$@" in a for loop and save the file.**

**Screenshot: Q6\_script\_forloop\_vim.png**

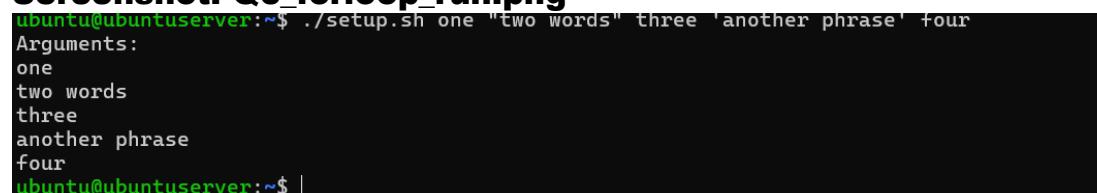


```
#!/bin/bash

echo "Arguments:"
for arg in "$@"; do
    echo "$arg"
done
```

**Run the script with mixed single and quoted multi-word arguments and capture the output showing each argument on its own line.**

**Screenshot: Q6\_forloop\_run.png**



```
ubuntu@ubuntuserver:~$ ./setup.sh one "two words" three 'another phrase' four
Arguments:
one
two words
three
another phrase
four
ubuntu@ubuntuserver:~$ |
```

---

## 7. Script: While Loop Summation and Functions

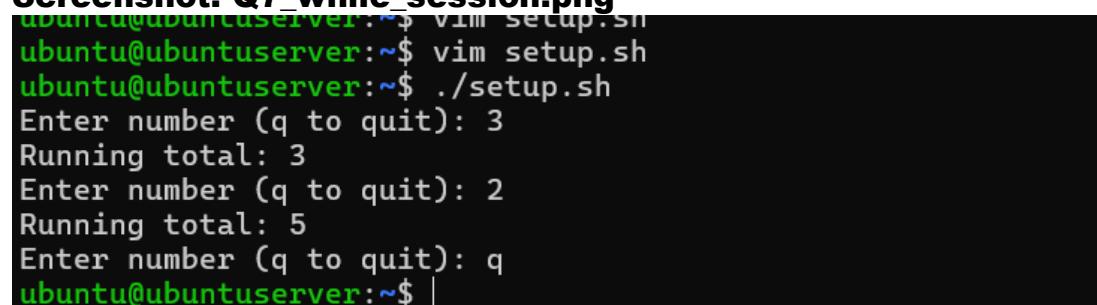
**Scenario:**

**Implement an interactive or non-interactive summation function and a demonstrated function that returns a numeric result.**

**Steps:**

**Write an interactive while-loop that accumulates numbers until q is entered and shows running totals.**

**Screenshot: Q7\_while\_session.png**



```
ubuntu@ubuntuserver:~$ vim setup.sh
ubuntu@ubuntuserver:~$ vim setup.sh
ubuntu@ubuntuserver:~$ ./setup.sh
Enter number (q to quit): 3
Running total: 3
Enter number (q to quit): 2
Running total: 5
Enter number (q to quit): q
ubuntu@ubuntuserver:~$ |
```

**Add a function that accepts two numeric arguments, returns their sum, and demonstrate capturing its result in a variable.**

**Screenshot: Q7\_function\_sum.png**

```
ubuntu@ubuntuserver:~$ vim setup.sh
ubuntu@ubuntuserver:~$ ./setup.sh
Sum from function: 10
ubuntu@ubuntuserver:~$ |
```