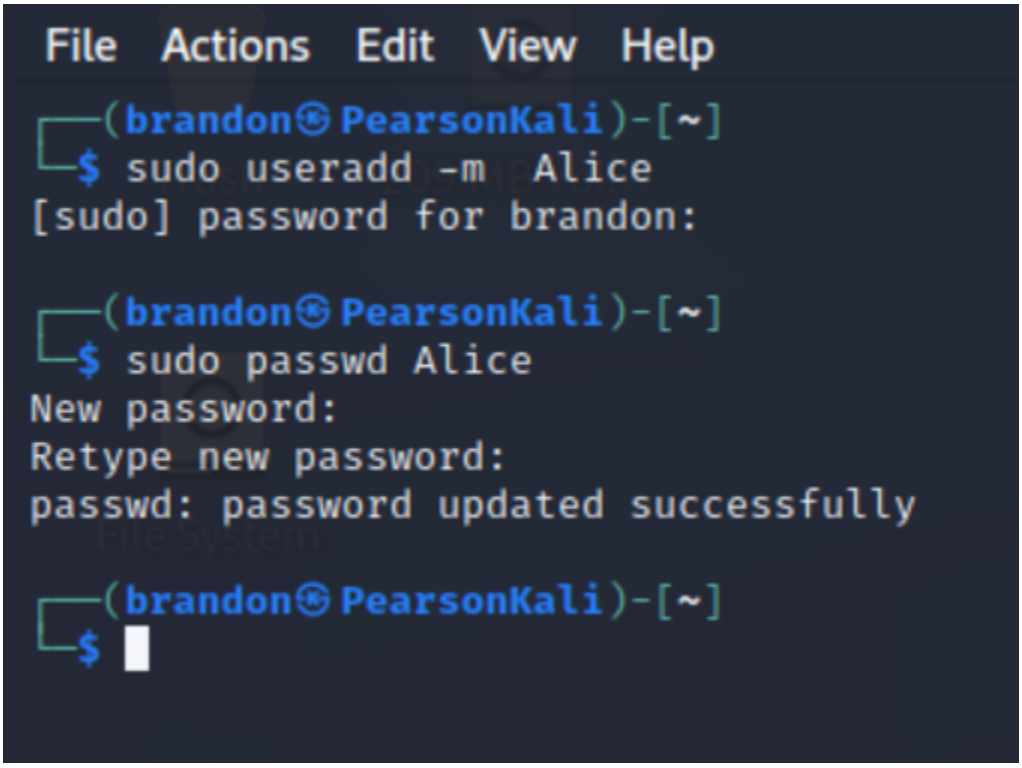
Task A - Backup your system (Using crontab) [100 points]
Scenario: Performing system backup can be time-consuming, and the process is often overlooked. For this scenario,
1. (10 Points) Create a new user Alice (with home directory) and



2. (50 Points) Write a shell script that backups Alice's home directory by creating a tar file
(tape archive), using the following steps:
i. Take 2 inputs with their values- your MIDAS name and current date (for example, midas=svatsa).
ii. Create a variable named as filename that should be assigned the value as
MIDAS-date (example output after executing the script would be like,

svatsa-2021.3.17-01.16.430).

```
#!/bin/bash

midas=bpear
current_date=$(date +%Y.%m.%d.-%H.%M.%S)
tar_file=$midas-$current_date

echo $tar_file

┌──(brandon㉿PearsonKali)-[~]
└─$ vi script9.sh

┌──(brandon㉿PearsonKali)-[~]
└─$ chmod a+x script9.sh

┌──(brandon㉿PearsonKali)-[~]
└─$ ls -l script9.sh
-rwxr-xr-x 1 brandon brandon 113 Mar 28 19:36 script9.sh
```

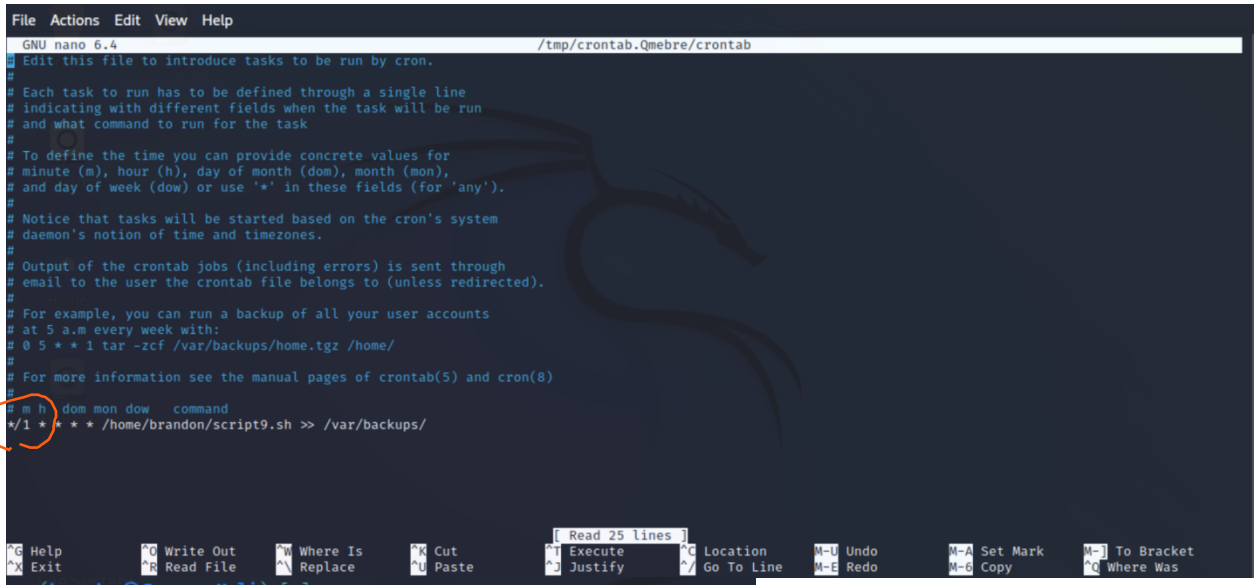
iii. Using tar command, create a tape archive for Alice's home directory (/home/Alice) and the filename created above (in step-2-ii). (Please learn about tar command in Linux for its usage)

b. Move the tape archive file/tar file (created in step 2-iii) to /var/backups/ directory using correct command in linux.
c. To optimize the disk usage, pick a compression algorithm (bz2, gzip, or xv) to
compress the tar file you created in /var/backups/ in the previous step-2b.

```
#!/bin/bash

midas=bpear
current_date=$(date +%Y.%m.%d.-%H.%M.%S)
tar_file='/var/backups/'  $midas-$current_date
gzip /var/backups/"$tar_file.tar"

echo $tar_file
```

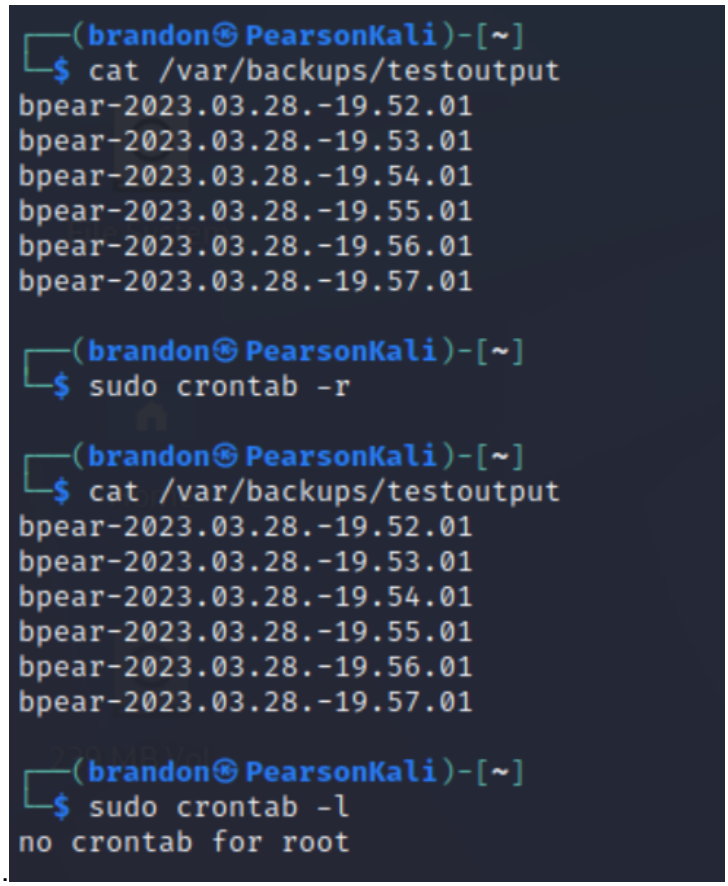
3. (30 Points) Create a crontab file to keep the scheduled task running for 3 minutes, then

check the contents in the /var/backups directory. Your output should be look similar
to the following:




```
┌──(brandon㉿PearsonKali)-[~]
└─$ ls /var/backups/
apt.extended_states.0  apt.extended_states.1.gz

┌──(brandon㉿PearsonKali)-[~]
└─$ ls /var/backups/
apt.extended_states.0  apt.extended_states.1.gz  testoutput

┌──(brandon㉿PearsonKali)-[~]
└─$ cat /var/backups/testoutput
bpear-2023.03.28.-19.52.01

┌──(brandon㉿PearsonKali)-[~]
└─$ cat /var/backups/testoutput
bpear-2023.03.28.-19.52.01
bpear-2023.03.28.-19.53.01

┌──(brandon㉿PearsonKali)-[~]
└─$ cat /var/backups/testoutput
bpear-2023.03.28.-19.52.01
bpear-2023.03.28.-19.53.01
bpear-2023.03.28.-19.54.01
```

4. (10 Points) Cancel the crontab jobs


┌──(brandon㉿PearsonKali)-[~]
└─$ cat /var/backups/testoutput
bpear-2023.03.28.-19.52.01
bpear-2023.03.28.-19.53.01
bpear-2023.03.28.-19.54.01
bpear-2023.03.28.-19.55.01
bpear-2023.03.28.-19.56.01
bpear-2023.03.28.-19.57.01
┌──(brandon㉿PearsonKali)-[~]
└─$ sudo crontab -r
┌──(brandon㉿PearsonKali)-[~]
└─$ cat /var/backups/testoutput
bpear-2023.03.28.-19.52.01
bpear-2023.03.28.-19.53.01
bpear-2023.03.28.-19.54.01
bpear-2023.03.28.-19.55.01
bpear-2023.03.28.-19.56.01
bpear-2023.03.28.-19.57.01
┌──(brandon㉿PearsonKali)-[~]
└─$ sudo crontab -l
no crontab for root