

Surname, Name:

Section:

Student No:

**Closed Book, closed note exam. You are required to write down commands with necessary arguments and options; and make sure that they work. Your script and output should match. Give the best result that you can give!**

**Each question worths 4 points unless otherwise stated.**

**Over 100 points is bonus.**

**Unless otherwise stated for question k, your answers as command must be in k.sh and output should in k.txt both should be in Answers directory.**

SIGNATURE .....

Time of Submission:

**Prelude:** Before solving questions you should:

- let NAME be your FirstLast name as ascii (MAkgul, ASOzgur, LMessi, LionelMessi) (do not use Turkish Characters, ı ğ ş ç ü ö )
- create NAME and NAME/Answers directories `mkdir -p ~/NAME/Answers`
- script NAME/Answers/NAME.Log
- touch NAME/Answers/Your-Full-Name
- download the questions file and unzip it in NAME Directory, ( maintaining directory structure), unzip file-path  
`cd ~/NAME` or simply `cd NAME`  
use one of the commands  
`wget http://liste.ctis.bilkent.edu.tr/courses/166/LabM2.zip`  
`wget http://lab7t/download/LabM2.zip`

unzip LabM2

- `mkdir -p ~/NAME/Answers/Dir{1,2,3,4,5,6,7,8,9}`
- `export PATH=.:$PATH`
- **Your working directory will be LabQ, files that you operate on will be in LabQ. Your answers will be written under NAME; shell scripts and solution files under NAME/Answers; you need to redirect selected output to NAME/Answers directory. Also commands that you use should ve saved/copied under Answers. You can collect commands under cevaplar.sh provided you include question numbers**

**For some questions, you may work directly in Name/Answers. Please indicate that in cevaplar.sh.**

**Remember final versions of all \*.sh Awk, Sed and \*.SH files will be in NAME/Answers.**

**Your scripts should work.**

- **When you finish** (that is when exam ends), you will zip NAME directory with command  
`cd ; zip -ry NAME NAME`
- upload NAME.zip

1. copy everything in LabQ into Dir1 with cp . Keep owner and date info. term LabQ will not be copied but everything inside it (recursively). Use relative addressing

```
cp -Ra . ../Answers/Dir1
```

2. copy everything in the current directory into Dir2 using tar without using any explicit tar file

```
tar cf - . | (cd ../Answers/Dir2; tar xf - ) OR
tar cf - . | tar xf - -C ../Answers/Dir2
```

3. copy everything in the current directory into Dir3 using tar via creating tar file LabQ.tar

```
tar cf ../LabQ.tar . ; tar xf ../LabQ.tar -C ../Answers/Dir3
OR tar cf ../LabQ.tar . ; cd ../Answers/Dir3 : tar xf ../../LabQ.tar
```

4. copy everything in the current directory into Dir4 using rsync

```
rsync -av . ../Answers/Dir4
```

5. copy everything in the current directory into Dir5 using a cpio file

```
( find . | cpio -o > ../LabQ.cpio; cd ../Answers/Dir5; cpio -id < ../../LabQ.cpio) OR
find . | cpio -o > ../LabQ.cpio; cd ../Answers/Dir5; cpio -id < ../../LabQ.cpio ; cd
```

6. copy all \*.txt and \*.JPEG files in LabQ (on the surface and within directories) into Dir6 using cpio

```
find . -type f \( -name "*.txt" -o -name "*.JPEG" \) | cpio -pd ../Answers/Dir6 OR
find . -type f -name "*.txt" | cpio -pd ../Answers/Dir6 AND
find . -type f -name "*.JPEG" | cpio -pd ../Answers/Dir6
%vfil
```

7. combine all \*.txt files within LabQ hierarchy which are more than 5k and less than 20k in size into file ALL.TXT in Answers

```
find . -type f -name "*.txt" -size +5k -size -20k | xargs cat > ../Answers/ALL.TXT
```

8. Consider all files in the current directory: including sub directories  
- determine all lines among all files which contains words ayse or elif, case insensitive

```
egrep -ihr "elif|ayse" . > ../Answers/8.txt
```

9. Consider all \*.txt files in LabQ (on the surface) determine files which contains words **elif** or **ayse** case insensitive on a line, (under 9.txt) and combine them (under 9.TXT)

```
egrep -il "elif|ayse" *.txt > ../Answers/9.txt
egrep -il "elif|ayse" *.txt | xargs cat > ../Answers/9.TXT
```

10. find all \*.txt files and \*.SH within LabQ hierarchy and put these files in TXT1.zip

```
find . -type f \( -name "*.txt" -o -name "*.SH" \) | xargs zip ../Answers/TXT1
OR find . -type f -name "*.txt" | xargs zip ../Answers/TXT1
AND find . -type f -name "*.SH" | xargs zip -u ../Answers/TXT1
```

11. find all \*.txt files within LabQ hierarchy and put these files in TXT1.cpio

```
find . -type f -name "*.txt" | cpio -o > ../Answers/TXT1.cpio
```

12. extract contents of AB.tar.gz into Dir7 so that after the operation AB.tar.gz remains intact .

```
tar xzf AB.tar.gz -C ../Answers/Dir7
OR zcat AB.tar.gz | tar xf - -C ../Answers/Dir7
OR ( cd ../Answers/Dir7; tar xzf ../../LabQ/AB.tar.gz )
```

13. find lines in Dene3.txt.bz2 containing words elif and ayse case insensitive without using bunzip2 and bzcata

```
bzgrep -iw elif Dene3.txt.bz2 | grep -iw ayse > ../Answers/13.txt
```

14. Combine Dene1.txt.Z Dene2.txt.gz and Dene3.txt.bz2 into DENE.txt so that after the operation original files remains intact. (n other words you can not use gunzip and gzip, or something similarly]

```
zcat Dene1.txt.Z > ../Answers/DENE.txt
gzcat Dene2.txt.gz >> ../Answers/DENE.txt
bzcat Dene3.txt.gz >> ../Answers/DENE.txt
```

15. give permissions of **yap.SH** as suid to owner ; owner has read, write, execute; and read execute to group and others have write and execute rights. Use octal method

```
cp yap.sh ../Answers; chmod 04753 ../Answers/yap.SH
```

16. add read permissions for file **exec.SH** to all , add write and sticky bit to others, add to write and suid property to group; and execute to owner; all in a single statement, and use symbolic method

```
cp exec.SH ../Answers; chmod a+r,o+wt,g+ws,u+x ../Answers/exec.SH
```

17. set following permissions for **x.SH** using symbolic method: user read, write and execute, group read and execute and suid; and for others execute and write and sticky bit.

```
cp x.SH ../Answers/; chmod u=rwx,g=rxs,o=wxt ../Answers/x.SH
```

18. Given a.txt, we want to determine the lines containing words Net and Fox case sensitive and put them in files 18-1.txt, 18-2.txt and 18-3.txt . How would you do it using:(each **2**)

```
grep: grep -w Net a.txt | grep -w Fox > ../Answers/18-1.txt
sed: sed -n '/\<Net\>/p' a.txt | sed -n '/\<Fox\>/p' > ../Answers/18-2.txt
awk: awk '/\<Net\>/' a.txt | awk '/\<Fox\>/' > ../Answers/18-3.txt
```

19. given a.txt, we want to determine the lines containing only one of strings Net and Fox and put them in files 19-1.txt, 19-2.txt and 19-3.txt How would you do it using:(each **2**)

```
grep: grep Net a.txt | grep -v Fox > ../Answers/19-1.txt
      grep -v Net a.txt | grep Fox >> ../Answers/19-1.txt
sed: sed -n '/Net/p' a.txt | sed '/Fox/d' > ../Answers/19-2.txt
      sed -n '/Fox/p' a.txt | sed '/Net/d' >> ../Answers/19-2.txt
awk: awk '/Net/' a.txt | awk '! /Fox/' > ../Answers/19-3.txt
      awk '! /Net/' a.txt | awk '/Fox/' >> ../Answers/19-3.txt
```

20. Given A.txt we want to: (each **2** points )

- Delete all strings Net and Fox (using **Sed** )

Give command and write new file as Answers/B1.txt

```
sed 's/Net//g' A.txt | sed 's/Fox//g' > ../Answers/B1.txt
```

- On lines containing Internet, replace Fox with Firefox

Give command and write file as Answers/B2.txt (using **vi/vim**)

```
vi/vim: :g/Internet/s/Fox/Firefox/g, :w ../Answers/B2.txt ; :q!
```

- Do the above using sed, and write into B3

```
sed '/Internet/s/Fox/Firefox/g' A.txt > ../Answers/B3.txt
```

- On lines 1 to line containing ElifNet insert "BASLA " at the beginning, Give commands and write file as Answers/B4.txt

```
sed '1,/ElifNet/s/^/BASLA / ' A.txt > ../Answers/B4.txt
```

- On line starting with "minternet" to line with BitNeT add to the end of each line word " BITTI" Give commands and write file as Answers/B5.txt

```
sed '/minternet/,/Bitnet/s/$/ BITTI/' A.txt > ../Answers/B5.txt
```

21. given satis.txt, compute total number of units sold for each fruit using Awk. Hint: use associative arrays. Print amounts for apple and orange.

awk -f ../Answers/Awk.21 satis.txt > ../Answers/21.txt with Awk.21 as:

```
{sum[$1]+=$2}
END{ print "results:"
      print "apples: ", sum["apple"]
      print "oranges", sum["orange"]
}
```

22. we want to run /usr/bin/temizlik.sh every Sunday and Thursday 2:25 in January, March, May and August. How would you do it ?

crontab entry: 25 2 \* 1,3,5,8 0,4 /usr/bin/temizlik.sh via crontab -e

23. we want run long-jobs.sh at 2:00 three days later . How would you do it ?

at -f long-jobs.sh 2:00 + 3 days

24. How would you determine all httpd processes and kill it ?

```
ps ax | grep httpd | awk '{ print $1 }' | xargs kill -9
```

25. stat.c is a simple program to compute mean and variance. stat is a compiled program from stat.c. a sample input is given as input.txt. Write a simple shell script when executed, will run program stat with input input.txt and will write its output in output.txt , and possible errors to error.txt. Please run the resulting shell script and output.txt and error.txt should be in Answers as well as shell script

stat.sh:

```
#!/bin/sh
./stat < input.txt > ../Answers/output.txt 2 > ../Answers/errors.txt
```

run stat.sh as ./stat.sh

26. Determine list of public rsync archives(modules) at 139.179.33.200 and copy files in these modules into Answers

```
rsync -av 139.179.33.200::
```

```
rsync -av 139.179.33.200::dene ../Answers/
```

```
rsync -av 139.179.33.200::denek ../Answers/
```

27. In 139.179.33.200 there is user named **ctis** and passwd ctis1548. create an empty file with your Name.LastName (ascii only) and copy using rsync into /home/ctis

```
rsync -av Name.LastName ctis@139.179.33.200:/home/ctis/
ctis@139.179.33.200's password:
sending incremental file list
Name.LastName
```

```
sent 91 bytes received 31 bytes 14.35 bytes/sec
total size is 0 speedup is 0.00
```

28. In the above machine there is an unlisted module **deney** ) with user ctisx is authorized to upload into it. user ctisx has the password ctisx. Upload Name.LastName into module deney using rsync.

```
rsync -av Name.LastName ctisx@139.179.33.200::deney
Password:
sending incremental file list
Name.LastName

sent 87 bytes received 27 bytes 17.54 bytes/sec
total size is 0 speedup is 0.00
```

29. Write bash script which will print:

- "Welcome"
- Name of script
- Number of arguments
- Each argument on a line by itself
- "Goodbye"

./29.sh 1 3 5 elif ayse "fatma hasan " > ../Answers/29.tx where 29.sh is

```
#!/bin/sh
echo "Welcome"
echo "My Name: " $0
echo " Number of Arguments: " $#
for x
do
    echo $x
done
echo " Goodbye"
exit 0
```

30. Write a bash script which will sum its arguments (assuming all integers) and

- print argument vector at the beginning
- at the end print " Sum of x terms is equal to y" where x is the number of arguments and y is total
- use while and shift

./30.sh 1 2 3 4 5 > ../Answers/30.txt where 30.sh is:

```
#!/bin/sh
echo $@
sayi=$#

sum=0
while [ $# -ge 1 ]
do
    sum=$(( sum + $1 ))
    shift
done

echo "Sum of $sayi terms is equal to: " $sum
```

31. Given grade.txt, determine number of A, F, and number of passing students. use Bash script with case construct.

./31.sh > ../Answers/31.txt where 31.sh is:

```
#!/bin/sh

while read isim not
do
((ns++))
case $not in
  A) ((nA++))
      ;;
  F) ((nF++))
      ;;
  *) ;;
esac
done < grade.txt
gecen=$(( ns - nF )
echo " Num of A's" $nA
echo "Number of F's " $nF
echo "Dersi Gecen ogrenci: " $gecen
exit 0
```

32. Write a bash script which will take an arbitrary number of arguments; for each arguments, bash script will do the following:

- i) check whether arguments exists as an object of current system, if it is not give an error message and continue (exit with value 1)
- ii) if its an ordinary file, print its size and exits
- iii) if its a directory print its disk usage and exit
- iv) Otherwise, print a message like " object x exists but it is not an ordinary file or directory" and exit . **8 pts**

./32.sh 32.sh /dev/log YOKYOK > ../Answers/32.txt is a sample run, where 32.sh is

```
#!/bin/sh
for x
do
nesne=$x
if [ ! -e $nesne ]
then
echo " $nesne does not exists"
continue
fi
if [ -f $nesne ]
then
set -- $( ls -l $nesne )
echo " size of ordinary file:" $nesne $5
elif [ -d $nesne ]
then
echo "Directory usage of $nesne is " $( du -s $nesne )
else
echo " Object $nesne exists, but it is not a file or directory"
fi
done
```

33. You are given uyu-1.sh uyu-2.sh uyu-3.sh and uyu-4.sh executable shell scripts. **6 points**
- Run uyu-1.sh, uyu2-sh, uyu-3.sh and uyu-4.sh in the background successively, give the result of "jobs" in 33-1.txt in Answers.
  - put uyu-2.sh in foreground, and stop it, give results of "jobs" in33-2.txt in Answers
  - kill uyu-1.sh using % notation, and put uyu-2.sh (uyu-3.sh) into background and kill it, and give resulting "jobs" result in 33-3.txt

i) after running these:

```
./uyu-1.sh &
./uyu-2.sh &
./uyu-3.sh &
./uyu-4.sh &
```

Result of "jobs" is:

```
[1]  Running          ./uyu-1.sh &
[2]  Running          ./uyu-2.sh &
[3]- Running          ./uyu-3.sh &
[4]+ Running          ./uyu-4.sh &
```

ii)

```
LabQ fg %2
./uyu-2.sh
^Z
[2]+  Stopped          ./uyu-2.sh
LabQ$ jobs
[1]  Running          ./uyu-1.sh &
[2]+  Stopped          ./uyu-2.sh
[3]  Running          ./uyu-3.sh &
[4]-  Running          ./uyu-4.sh &
```

iii)

```
kill -9 %1
bg %2
kill -9 %2

[3]-  Running          ./uyu-3.sh &
[4]+  Running          ./uyu-4.sh &
```