

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!

SIGNATURE

.....

Time of Submission:

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

- use: `history -c`
- Your working directory will be `Name`, files that you operate on will be in `Name`. Your answers will be written under `NAME`.

When you finish:

```
use: history > NAME/History.txt
```

you will zip `NAME` directory with command

```
cd ; zip -r NAME NAME
```

- upload `NAME.zip`

**You need to have `History.txt`, `Name.Log` and `Cevaplar` as separate bash script files, which are checked and run. Make sure it works correctly**

**If not you would loose most of the credits**

**Consider the last digit of your student ID, and let it be `K`.**

**Then let `k=1` if `K=0, 3, 6, 9`; `k=2` if `K=1, 4, 7` and `k=3` if `K=2, 5, 8`**

**you will be working with `mail.k` otherwise stated.**

These are most used formats dealing with infile (input file), outfile (output file), shell variables in out

```
cat infile | commands > outfile
commands infile > outfile
commands < infile > outfile
echo $in | commands > outfile
out=$( echo $in | commands )
out=$( cat infile |commands )
```

Consider a conference system with user "chair" is the Chairman of Conference. There are authors and Reviewers. The file **tablo** contains lines of the form:

```
id:Rev1info;Rev2info;Rev3info
```

We may have 1 Reviewer or many. Each Reviewer-info contains e-mail address and name which is separated by ";". and space in name is replaced by - .

When authors uploads a file, system send a mail to chair as shown in mail.1, mail.2, and mail.3 files. we want to write a bash script which will mail to each reviewer an e-mail saying that "paper with id such-and-such is uploaded"

**Your task is to write such a script and place in mail system.**

You might do the following steps and combine them in a bash script.

**this was quiz that we solved in class. We will have similar small questions about the same problem and similar concepts**

You can not use any editor within scripts (such as nano, vi etc). Because Scripts will work without human intervention. You can use any program that works on standard input of file . You can use sed, awk, grep etc. in tablo, number of reviewers can change in each line, and size of header part of mail changes in each message.

**you can name shell files as 5b2.sh for question 5, b) section, second method**

1. select the correct line in mail.k using : (the line similar to **Submission ID 789 has been uploaded.**

a)grep, b) sed, c) awk ; write the output into 1a.txt, 1b.txt, 1c.txt

```
grep "^Submission" mail.k > 1a.txt or cat mail.k | grep ^Subm > 1a.txt
sed -n '/^Submission/p' mail.k > 1b.txt
awk '/^Submission/' mail.k > 1c.txt
```

b) just do it with grep on mail.0 and write onto 1d.txt

```
grep ^Submission Mail.0 > 1d.txt ("..." is optional)
```

2. instead of storing in a file, just store in a shell variable e1, f1 (just choose one of grep, sed or awk) and work mail.k and mail.0. (**I must see result of**

```
echo "E $e1"
```

**and** echo "F \$f1" ) in History, Log and Cevaplar

```
e1=$(grep ^Submission mail.k); echo "F $e1"
```

```
f1=$(cat mail.0 | grep ^Submission ); echo "F $f1"
```

3. a) Given 1a.txt, how would you obtain Submission ID (say 8, 28, 310), as id, give more than one method if you can

```
id=$( awk '{print $3}' 1a.txt ),or
```

```
id=$(cat 1a.txt |awk '{print $3}' )
```

```
id=$(sed -f SED.1 1a.txt) , where SED.1 is:
```

```
s/^Submission ID //
```

```
s/ has been uploaded.// (end of SED.1)
```

```
id=$(sed -e 's/Submission ID //' -e 's/ has been uploaded.//' 1a.txt)
```

b) how would you obtain from e1

```
id=$(echo $e1 | awk '{print $3 }') OR id=$(echo $e1 | sed -f SED.1)
```

4. a) How would you obtain Submission ID from mail.k in a single step via **awk** (no pipe is allowed). write the result in a4.txt,

```
awk '/^Submission/ {print $3 }' mail.k > a4.txt
```

b) How would you store in shell variable a4

```
a4=$(awk '/^Submission/ {print $3 }' mail.k
```

5. How would you obtain from mail.k files header.k and body.k ? Header is the beginning part of mail message including first empty line, and body is the rest. Save the resulting files as header.txt and body.txt

```
sed -n '/Return-Path:/,/^$/p' mail.k > header.k
```

```
sed -n '1,/^$/p' mail.k > header.k
```

```
sed '/^$/d' mail.k > header.k
```

```
sed -n '/^$/, $p' mail.k > body.k
```

```
sed '/Return-Path:/,/^$/d' mail.k > body.k
```

(Difference of empty line is acceptable)

6. How would you determine that there is no Submission ID. In other words:  
 a) how would you determine whether a file is empty? Give as many methods as you can (you can test on files bos.txt and dolu.txt)

i) `sayi=$(wc -l file.txt ) sayi=0` means file is empty, `sayi > 0` is non-empty

ii) `sayi=$(grep -c "." file.txt)`

As above; you can use `^` or `$` instead of `."`

iii) `set -- $(ls -l file.txt), echo $5 ; $5` is size of the file

iv) `set -- $(du -sh file.txt) ; echo $1 ; $1` is size of the file

v) `find file.txt -empty` returns file.txt if empty

`find . -type f -name file.txt -empty` will return `./file.txt` if empty, otherwise it will return NULL or nothing

- b) how would test whether shell variable id is empty

```
if [ $id ]; then; echo " $id is not empty"
else; echo " $id is empty"
fi
```

```
if [ -z $id ]; then; echo " $id is empty; fi
```

7. If there is no submission id (as in mail.0), how would you finish the process with a message to standard output (exit 0); that is test id for emptiness if yes exit the process, if not empty follow the next line (for the small shell file echo " ID is (its value) "

```
if [ -z $id ]; then
echo "id is empty"
exit 0
else
echo " ID is $id "
fi
```

8. select the line from "tablo" containing "id" found above and pick collection of Reviewer-info's in file rev-all.txt Do it with alternative ways if possible (rev-all-1.txt, rev-all-2.txt etc)

```
grep "^$id:" tablo | awk -F: '{print $2}' > rev-all-1.txt
sed -n '/^$id:/p' tablo | sed 's/^$id://' > rev-all-2.txt
awk -F: '/^$id:/ {print $2}' tablo > rev-all-3.txt
```

9. identify each Reviewerinfo which is separated by ";" , in a temporary file , say, revs.txt, each line contains different reviewers

```
SED.2      SED.3      AWK
s/;/\      s/,/\      BEGIN{FS=";"; OFS="\n"}
/g        /g        {for (i=1; i<=NF; i++) {print $i}}
```

- a) `sed -f SED.2 rev-all.txt > revs.txt`  
 b) `rev-all.txt | tr ';' '\n' > revs.txt`  
 c) `awk -f AWK rev-all.txt > revs.txt`  
 d) `A=( $(cat rev-all.txt | sed 's;/ /g' ) )` places entries in array A and  
`for j in ${A[@]}; do ; echo $j; done > revs.txt`

10. separate each Reviewerinfo as bash variables address, and name; and obtain AdSoyad which is name where it contains space " " instead of "-", i.e replace -'s with " " and for each reviewer; echo AdSoyad and address, ( do as many different methods as you can )

```
for i in $(cat revs.txt)
do
adres=$(echo $i | sed 's/,.*$//' )
name=$(echo $i | sed 's/^.*, //' )
AdSoyad=$(echo $name | sed 's/-/ /g')
done
```

```
for i in $(cat revs.txt)
do
adres=$( echo $i | awk -F, '{print $1 }' )
name=$( echo $i | awk -F, '{print $2 }' )
AdSoyad=$(echo $name | sed 's/-/ /g' )
done
```

```
for i in $(cat revs.txt)
do
B=( $(echo $i | sed 's/,/ /g' ) )
adres=${B[0]}
name=${B[0]}
AdSoyad=$(echo $name | sed 's/-/ /g' )
done
```

11. given address and AdSoyad ( you can set them for this part as you wish)

send a message as:

```
To: address
Subject: File uploaded
```

```
Dear AdSoyad,
Author of submission identified by id has been upload
Regards
The Chair
```

(id will be what you found, using shell variable id )

```
mailx -s"File Uploaded" <<BITTI
Dear $AdSoyad,
Author of submission identified by ID $id has been uploaded
Regards
The chair
BITTI
```

12. Suppose you wrote and placed the above script as /usr/bin/myyap, how would you activate this script ? which file do you work, and what do you write in it ?, which additional command you issue ? (total 15 points)

```
chmod +x /usr/bin/myyap
in aliases file:
chair: \chair .... "|/usr/bin/myyap
followed by newaliases command
```

You may use several shell variables, small scripts or auxiliary files. For each incoming mail you may use temporary files in /tmp. Recall that \$\$ gives process id of the running process, so that you can use files such as \$. \$\$.

Each part worth 7 points if not specified.