Lab (3)

1. Using vi write your CV in the file mycv. Your CV should include your name, age, school, college, experience,…

```
[amr@localhost ~]$ touch mycv
[amr@localhost ~]$ vi mycv
[amr@localhost ~]$
```

```
name : Amr Hossam
age: 27
EDUCATION: Faculty Of Engineering Computer and Systems Department
Phone number: 01092148480
Title: DevOps Engineer
```

2. Open mycv file using vi command then: Without using arrows state how to:

a. Move the cursor down one line at time.

j – Move the cursor down

b. Move the cursor up one line at time.

k – Move the cursor up

c. Search for word age

/age – search forward for age

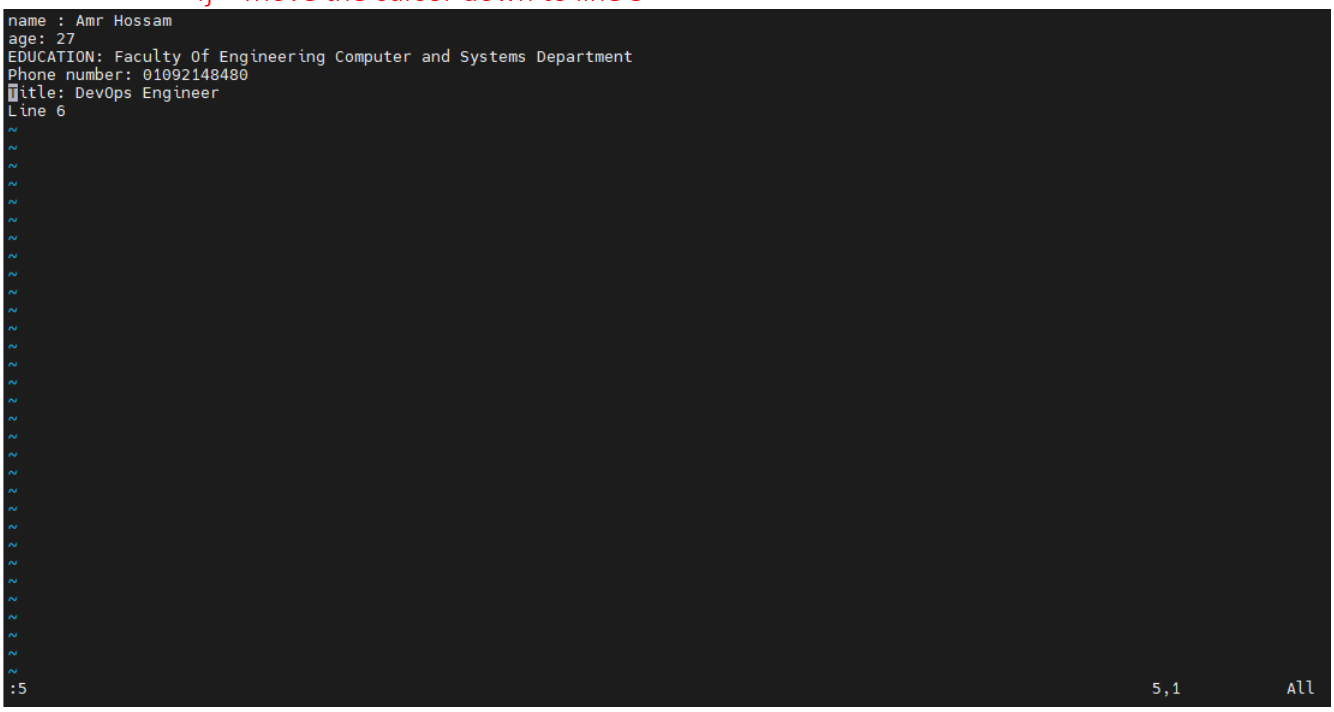
```
name : Amr Hossam
age: 27
EDUCATION: Faculty Of Engineering Computer and Systems Department
Phone number: 01092148480
Title: DevOps Engineer
/age
```

d. Step to line 5 (assuming that you are in line 1 and file is more than 5 lines).

:5– Move the cursor down to line 5

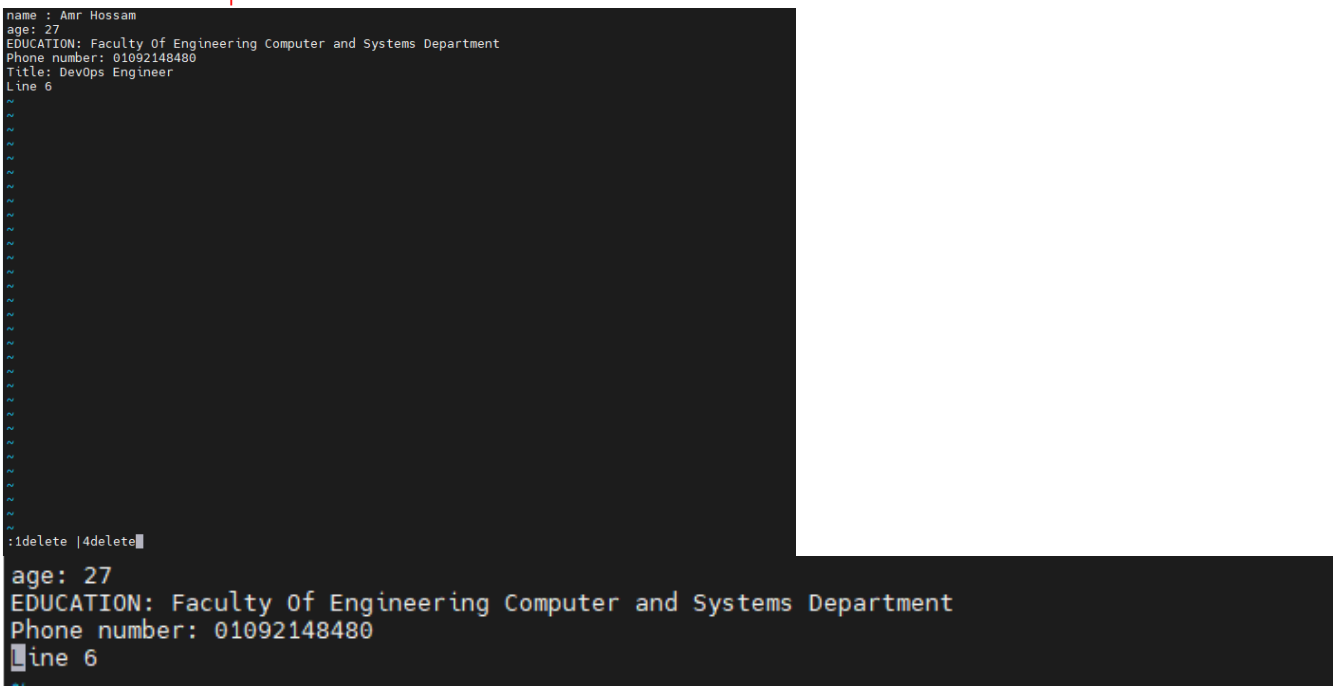
or

4j – Move the cursor down to line 5



e. Delete the line you are on and line 5.

:1 delete |4 delete to delete line 1 and line 5



f. How to step to the end of line and change to writing mode in one-step.

A – Move the cursor to the end of line and change to writing mode in one-step

3. List the available shells in your system.

```
[amr@localhost ~]$ cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
[amr@localhost ~]$
```

4. List the environment variables in your current shell.

```
[amr@localhost ~]$ printenv
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca
=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;3
1:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=0
1;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*
.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=0
1;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01
;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35
:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*
.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;
35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.m4a=01;36:*.
mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.oga=01;36:*.opus=01;36:*.spx=01;36:*.xspf
=01;36:
```

5. List all of the environment variables for the bash shell.

```
[amr@localhost ~]$ declare -p
declare -- BASH="/bin/bash"
declare -r BASHOPTS="checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:force_fignore:histappend:interactive_comment
s:login_shell:progcomp:promptvars:sourcepath"
declare -ir BASHPID
declare -- BASHRCSOURCED="Y"
declare -A BASH_ALIASES=()
declare -a BASH_ARGC=()
declare -a BASH_ARGV=()
declare -A BASH_CMDS=()
declare -- BASH_COMMAND
declare -a BASH_COMPLETION_VERSINFO=([0]="2" [1]="7")
declare -a BASH_LINENO=()
declare -ar BASH_REMATCH=()
declare -a BASH_SOURCE=()
declare -- BASH_SUBSHELL
declare -ar BASH_VERSINFO=([0]="4" [1]="4" [2]="20" [3]="1" [4]="release" [5]="x86_64-redhat-linux-gnu")
declare -- BASH_VERSION="4.4.20(1)-release"
declare -- COLUMNS="137"
declare -- COMP_WORDBREAKS="
\"'><;|&(:"
declare -- CONF_FILE="/etc/vdoconf.yml"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:abstract=/tmp/dbus-EWubYfCJT5,guid=efb395301485dd432aba70d26437b1d1"
declare -- DBUS_SESSION_BUS_PID="2448"
```

6. What are the commands that list the value of a specific variable?

echo $VARIABLE

printenv VARIABLE

```
[amr@localhost ~]$ echo $SHELL
/bin/bash
```

```
[amr@localhost ~]$ printenv SHELL
/bin/bash
```

7. Display your current shell name.

```
[amr@localhost ~]$ echo $SHELL
/bin/bash
```

1. List the user commands and redirect the output to /tmp/commands.list

```
[amr@localhost ~]$ compgen -c > /tmp/commands.list
[amr@localhost ~]$ cat /tmp/commands.list
egrep
fgrep
grep
l.
ll
ls
vi
```

2. Count the number of user commands

```
[amr@localhost ~]$ wc -l /tmp/commands.list
2346 /tmp/commands.list
[amr@localhost ~]$
```

3. Get all the users names whose first character in their login is ‘g’.

```
[amr@localhost ~]$ getent passwd | awk -F: '$1 ~ /^g/ {print $1}'
games
geoclue
gluster
gdm
gnome-initial-setup
[amr@localhost ~]$
```

4. Get the logins name and full names (comment) of logins starts with “g”.

```
[amr@localhost ~]$ getent passwd | awk -F: '$1 ~ /^g/ {print $1,$5}'
games games
geoclue User for geoclue
gluster GlusterFS daemons
gdm
gnome-initial-setup
[amr@localhost ~]$
```

5. Save the output of the last command sorted by their full names in a file.

```
[amr@localhost ~]$ getent passwd | awk -F: '$1 ~ /^g/ {print $1,$5}' | sort -k2 >sorted_logins
[amr@localhost ~]$ cat sorted_logins
gdm
gnome-initial-setup
games games
gluster GlusterFS daemons
geoclue User for geoclue
[amr@localhost ~]$
```

6. Write two commands: first: to search for all files on the system that named .bash_profile. Second: sorts the output of ls command on / recursively, Saving their output and error in 2 different files and sending them to the background.

```
[amr@localhost ~]$ find / -name .bash_profile > output.txt 2> error-output.txt
[amr@localhost ~]$ ls -R / >output.txt 2>error-output.txt &
[2] 4402
```

7. Display the number of users who is logged now to the system.

```
[amr@localhost ~]$ who -q
amr amr
# users=2
[amr@localhost ~]$
```

8. Display lines 7 to line 10 of /etc/passwd file

```
[amr@localhost ~]$ sed -n '7,10p' /etc/passwd
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
[amr@localhost ~]$
```

9. What happens if you execute:

a. cat filename1 | cat filename2

filename2 only displayed on the terminal and file1 not printed to file2

```
[amr@localhost ~]$ cat filename1 | cat filename2
HI, File2 is here
```

b. ls | rm

This command deletes everything listed in the first command and not recommended to delete anything with this way

```
[amr@localhost ~]$ ls | rm
rm: missing operand
Try 'rm --help' for more information.
[amr@localhost ~]$
```

c. ls /etc/passwd | wc –l

As this command counts the number of line for the output of ls /ets/passwd and the output is listed in 1 line, so the output is 1

```
[amr@localhost ~]$ ls /etc/passwd | wc -l
1
```

10. Issue the command sleep 100.

```
[amr@localhost ~]$ sleep 100
```

11. Stop the last command.

```
[amr@localhost ~]$ sleep 100
^Z
[1]+  Stopped                 sleep 100
[amr@localhost ~]$
```

12. Resume the last command in the background

```
[amr@localhost ~]$ sleep 100
^Z
[1]+  Stopped                 sleep 100
[amr@localhost ~]$ bg %1
[1]+ sleep 100 &
[amr@localhost ~]$ jobs
[1]+  Running                 sleep 100 &
[amr@localhost ~]$
```

13. Issue the jobs command and see its output.

```
[amr@localhost ~]$ jobs
[1]+  Running                 sleep 100 &
[amr@localhost ~]$
```

14. Send the sleep command to the foreground and send it again to the background.

```
[amr@localhost ~]$ fg %1
sleep 100
^Z
[1]+  Stopped                 sleep 100
[amr@localhost ~]$ bg %1
[1]+ sleep 100 &
[amr@localhost ~]$
```

15. Kill the sleep command.

```
[amr@localhost ~]$ sleep 100 &
[1] 2503
[amr@localhost ~]$ kill 2503
[amr@localhost ~]$ jobs
[1]+  Terminated              sleep 100
[amr@localhost ~]$
```

16. Display your processes only

```
[amr@localhost ~]$ ps -u amr
    PID TTY          TIME CMD
   2302 ?        00:00:00 systemd
   2306 ?        00:00:00 (sd-pam)
   2323 ?        00:00:00 pulseaudio
   2325 ?        00:00:00 sshd
   2349 ?        00:00:00 sshd
   2359 ?        00:00:00 sftp-server
   2400 ?        00:00:00 dbus-daemon
   2405 pts/0    00:00:00 bash
   2448 ?        00:00:00 dbus-daemon
   2451 ?        00:00:00 dbus-kill-proce
   2459 ?        00:00:00 gio
   2464 ?        00:00:00 gvfsd
   2469 ?        00:00:00 gvfsd-fuse
   3666 pts/0    00:00:00 sleep
   3684 pts/0    00:00:00 ps
[amr@localhost ~]$
```

17. Display all processes except yours

```
[amr@localhost ~]$ ps aux | grep -v amr
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  1.7  0.2 241128 14216 ?        Ss   09:01   0:03 /usr/lib/systemd/systemd --switched-root --system --deserialize 18
root           2  0.0  0.0      0     0 ?        S    09:01   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   09:01   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   09:01   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I    09:01   0:00 [kworker/0:0-events]
root           6  0.0  0.0      0     0 ?        I<   09:01   0:00 [kworker/0:0H-events_highpri]
root           7  0.0  0.0      0     0 ?        I    09:01   0:00 [kworker/0:1-events_power_efficient]
root           8  0.0  0.0      0     0 ?        I    09:01   0:00 [kworker/u256:0-nvme-wq]
root           9  0.0  0.0      0     0 ?        I<   09:01   0:00 [mm_percpu_wq]
root          10  0.0  0.0      0     0 ?        S    09:01   0:00 [rcu_tasks_rude_]
root          11  0.0  0.0      0     0 ?        S    09:01   0:00 [rcu_tasks_trace]
root          12  0.0  0.0      0     0 ?        S    09:01   0:00 [ksoftirqd/0]
root          13  0.0  0.0      0     0 ?        I    09:01   0:00 [rcu_sched]
root          14  0.0  0.0      0     0 ?        S    09:01   0:00 [migration/0]
```

18. Use the pgrep command to list your processes only

```
[amr@localhost ~]$ pgrep -u amr
2302
2306
2323
2325
2349
2359
2400
2405
2448
2451
2459
2464
2469
3666
[amr@localhost ~]$
```

19. Kill your processes only.

```
[amr@localhost ~]$ pkill -u amr
```

Additional Questions

1- Which command gives an overview of all current shell jobs?

```
[amr@localhost ~]$ jobs
[1]+  Running                 sleep 2000 &
[amr@localhost ~]$
```

2- How do u stop the current shell job to continue running it in the background?

```
[amr@localhost ~]$ sleep 2000
^Z
[1]+  Stopped                 sleep 2000
[amr@localhost ~]$ bg %1
[1]+ sleep 2000 &
```

3- Which keystroke combination can u use to cancel the current shell job?

```
[amr@localhost ~]$ sleep 2000
^C
[1]+  Terminated              sleep 2000
[amr@localhost ~]$
```

4- A user is asking u to cancel one of the jobs he has started. You cannot access the shell that user currently is working from. What can u do to cancel his job anyway?

We can check his user process to get the PID of the job by using
ps –u “username”
then we can kill it using command
kill –SIGTERM “PID”