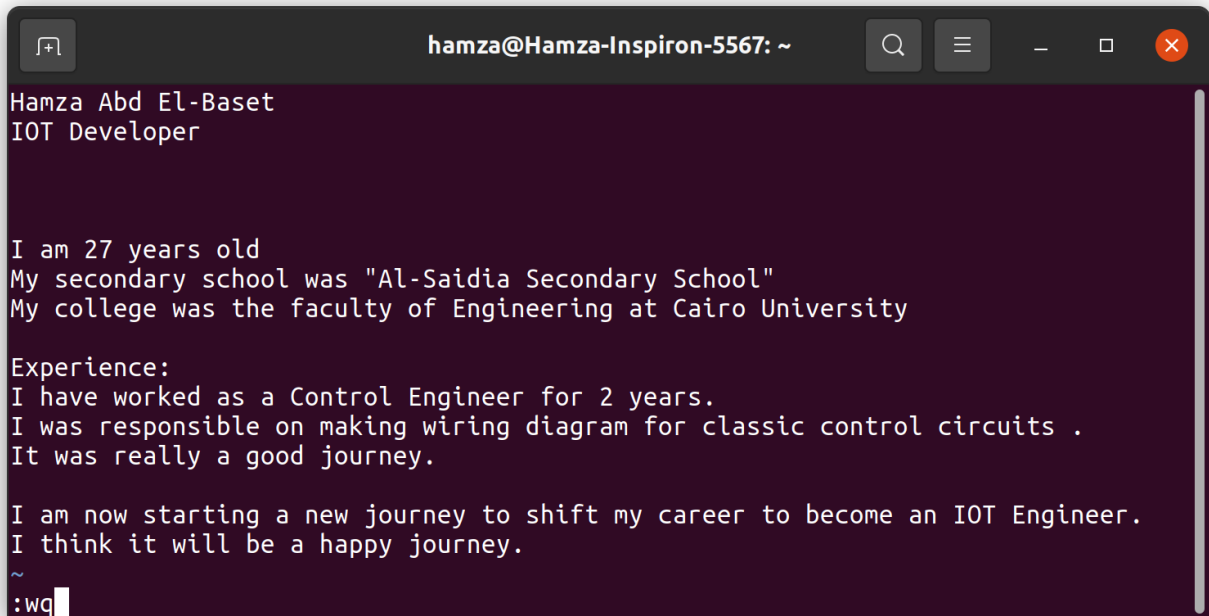


Lab 3

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

`touch mycv`
`vi mycv`



```
hamza@Hamza-Inspiron-5567: ~  
Hamza Abd El-Baset  
IOT Developer  
  
I am 27 years old  
My secondary school was "Al-Saidia Secondary School"  
My college was the faculty of Engineering at Cairo University  
  
Experience:  
I have worked as a Control Engineer for 2 years.  
I was responsible on making wiring diagram for classic control circuits .  
It was really a good journey.  
  
I am now starting a new journey to shift my career to become an IOT Engineer.  
I think it will be a happy journey.  
~  
:wq
```

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

a. Move the cursor down one line at time.

`Press j`

b. Move the cursor up one line at time.

`Press k`

c. Search for word age

`Write /age then press Enter`

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

`Write 5G`

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

`Press dd then :5d`

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

Press A

3. List the available shells in your system.



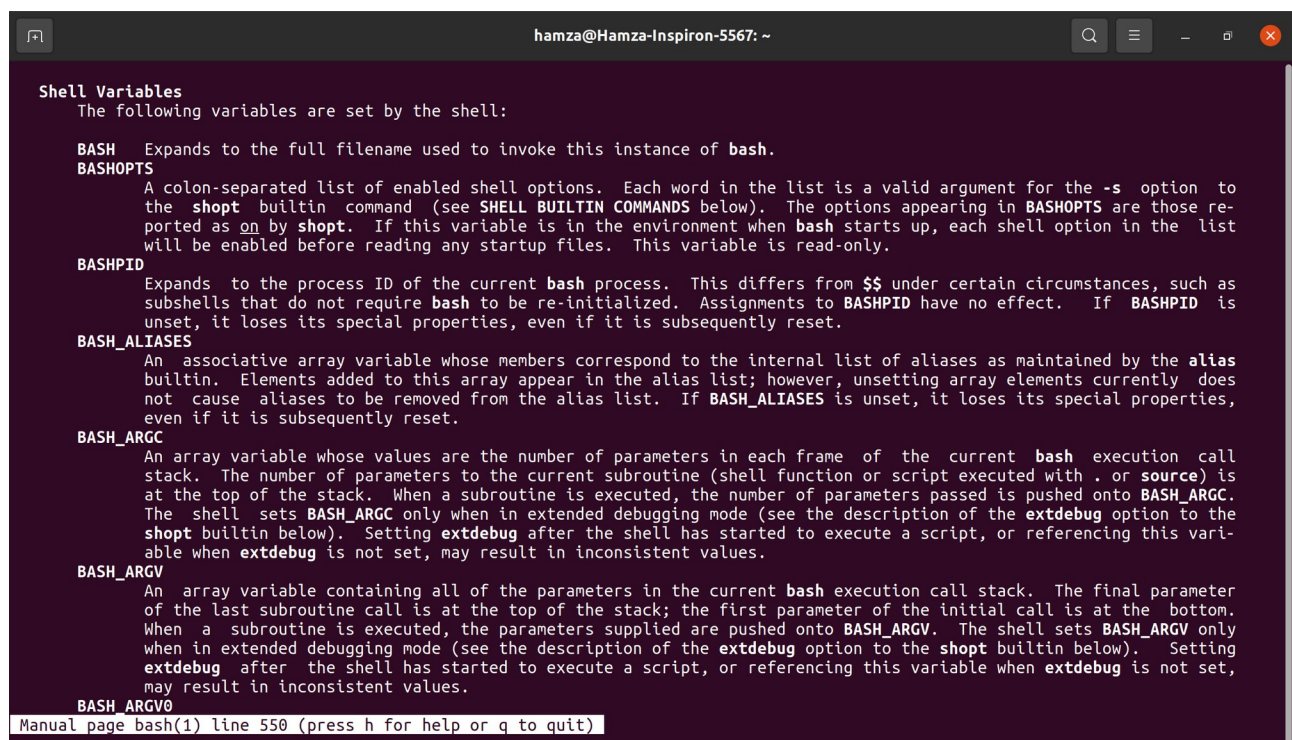
```
hamza@Hamza-Inspiron-5567: ~  
hamza@Hamza-Inspiron-5567:~$ cat /etc/shells  
# /etc/shells: valid login shells  
/bin/sh  
/bin/bash  
/usr/bin/bash  
/bin/rbash  
/usr/bin/rbash  
/bin/dash  
/usr/bin/dash  
hamza@Hamza-Inspiron-5567:~$
```

4. List the environment variables in your current shell.

env

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

Man bash



```
hamza@Hamza-Inspiron-5567: ~  
Shell Variables  
The following variables are set by the shell:  
  
BASH Expands to the full filename used to invoke this instance of bash.  
BASHOPTS  
A colon-separated list of enabled shell options. Each word in the list is a valid argument for the -s option to the shopt builtin command (see SHELL BUILTIN COMMANDS below). The options appearing in BASHOPTS are those reported as on by shopt. If this variable is in the environment when bash starts up, each shell option in the list will be enabled before reading any startup files. This variable is read-only.  
BASHPID  
Expands to the process ID of the current bash process. This differs from $$ under certain circumstances, such as subshells that do not require bash to be re-initialized. Assignments to BASHPID have no effect. If BASHPID is unset, it loses its special properties, even if it is subsequently reset.  
BASH_ALIASES  
An associative array variable whose members correspond to the internal list of aliases as maintained by the alias builtin. Elements added to this array appear in the alias list; however, unsetting array elements currently does not cause aliases to be removed from the alias list. If BASH_ALIASES is unset, it loses its special properties, even if it is subsequently reset.  
BASH_ARGC  
An array variable whose values are the number of parameters in each frame of the current bash execution call stack. The number of parameters to the current subroutine (shell function or script executed with . or source) is at the top of the stack. When a subroutine is executed, the number of parameters passed is pushed onto BASH_ARGC. The shell sets BASH_ARGC only when in extended debugging mode (see the description of the extdebug option to the shopt builtin below). Setting extdebug after the shell has started to execute a script, or referencing this variable when extdebug is not set, may result in inconsistent values.  
BASH_ARGV  
An array variable containing all of the parameters in the current bash execution call stack. The final parameter of the last subroutine call is at the top of the stack; the first parameter of the initial call is at the bottom. When a subroutine is executed, the parameters supplied are pushed onto BASH_ARGV. The shell sets BASH_ARGV only when in extended debugging mode (see the description of the extdebug option to the shopt builtin below). Setting extdebug after the shell has started to execute a script, or referencing this variable when extdebug is not set, may result in inconsistent values.  
BASH_ARGV0  
Manual page bash(1) line 550 (press h for help or q to quit)
```

```
hamza@Hamza-Inspiron-5567: ~  
BASH_CMDS  
An associative array variable whose members correspond to the internal hash table of commands as maintained by the  
hash builtin. Elements added to this array appear in the hash table; however, unsetting array elements currently  
does not cause command names to be removed from the hash table. If BASH_CMDS is unset, it loses its special prop-  
erties, even if it is subsequently reset.  
BASH_COMMAND  
The command currently being executed or about to be executed, unless the shell is executing a command as the re-  
sult of a trap, in which case it is the command executing at the time of the trap.  
BASH_EXECUTION_STRING  
The command argument to the -c invocation option.  
BASH_LINENO  
An array variable whose members are the line numbers in source files where each corresponding member of FUNCNAME  
was invoked. ${BASH_LINENO[i]} is the line number in the source file (${BASH_SOURCE[i+1]}) where ${FUNC-  
NAME[i]} was called (or ${BASH_LINENO[i-1]} if referenced within another shell function). Use LINENO to obtain  
the current line number.  
BASH_LOADABLES_PATH  
A colon-separated list of directories in which the shell looks for dynamically loadable builtins specified by the  
enable command.  
BASH_REMATCH  
An array variable whose members are assigned by the =~ binary operator to the [[ conditional command. The element  
with index 0 is the portion of the string matching the entire regular expression. The element with index n is the  
portion of the string matching the nth parenthesized subexpression. This variable is read-only.  
BASH_SOURCE  
An array variable whose members are the source filenames where the corresponding shell function names in the FUNC-  
NAME array variable are defined. The shell function ${FUNCNAME[i]} is defined in the file ${BASH_SOURCE[i]} and  
called from ${BASH_SOURCE[i+1]}.  
BASH_SUBSHELL  
Incremented by one within each subshell or subshell environment when the shell begins executing in that environ-  
ment. The initial value is 0.  
BASH_VERSION  
A readonly array variable whose members hold version information for this instance of bash. The values assigned  
to the array members are as follows:  
BASH_VERSION[0] The major version number (the release).  
BASH_VERSION[1] The minor version number (the version).  
Manual page bash(1) line 587 (press h for help or q to quit)
```

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

echo

7. Display your current shell name.

```
hamza@Hamza-Inspiron-5567: ~  
hamza@Hamza-Inspiron-5567:~$ echo $SHELL  
/bin/bash  
hamza@Hamza-Inspiron-5567:~$
```

8. State the initialization files of: sh, ksh, bash.

Sh shell:

- /etc/profile
- \$HOME/.profile


ksh shell:

- /etc/profile
- \$HOME/.profile
- \$HOME/.kshrc

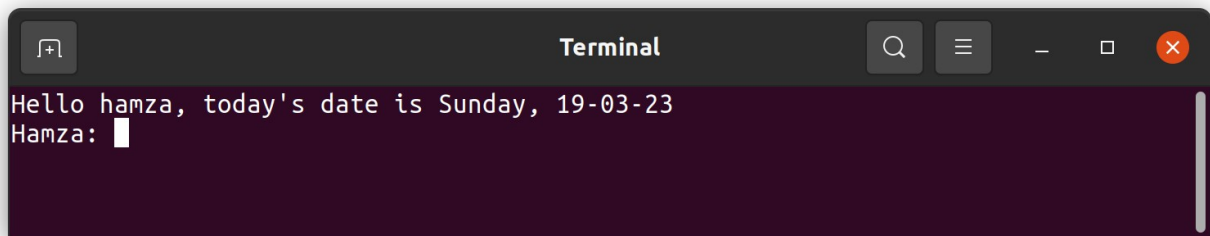
bash shell:

- `/etc/profile`
- `$HOME/.profile`
- `$HOME/.bashrc`

9. Edit in your profile to display date at login and change your prompt permanently.

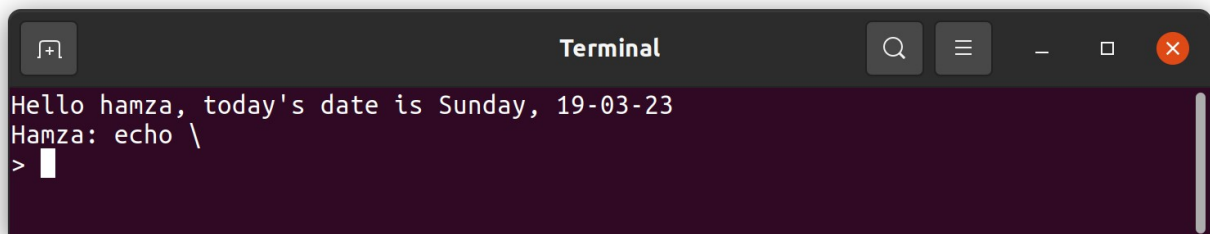


```
112 if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114 elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116 fi
117 fi
118
119 echo "Hello $USER, today's date is $(date +%A, %d-%m-%y)"
120 PS1='Hamza: '
121 |
```



```
Terminal
Hello hamza, today's date is Sunday, 19-03-23
Hamza: 
```

10. Execute the following command :
`echo \` then press enter



```
Terminal
Hello hamza, today's date is Sunday, 19-03-23
Hamza: echo \
> 
```

What is the purpose of `\` ?

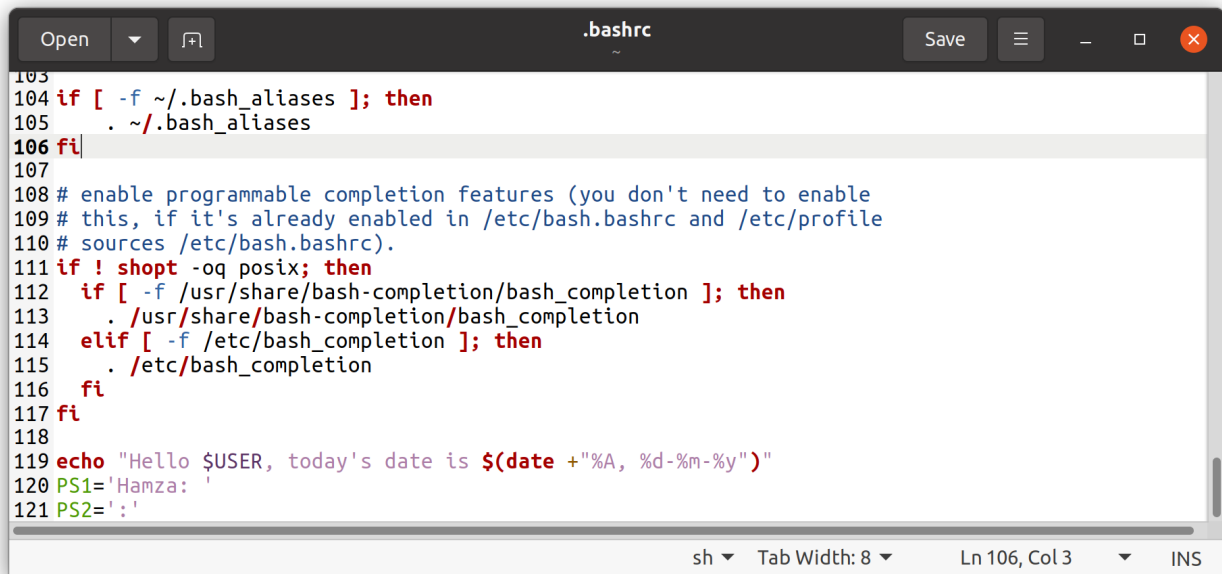
It escapes the next letter which is 'Enter' in our case

Notice the prompt `>` what is that?

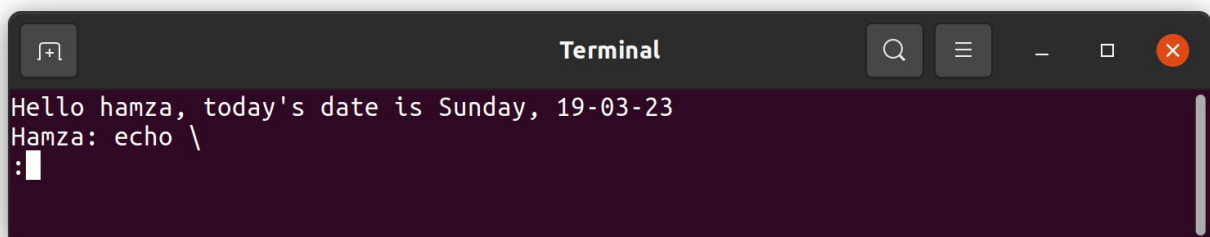
This prompt means the command isn't completed successfully
Enter must be typed again to complete the command.

and how can you change it from “>” to “:”.

`gedit .bashrc`

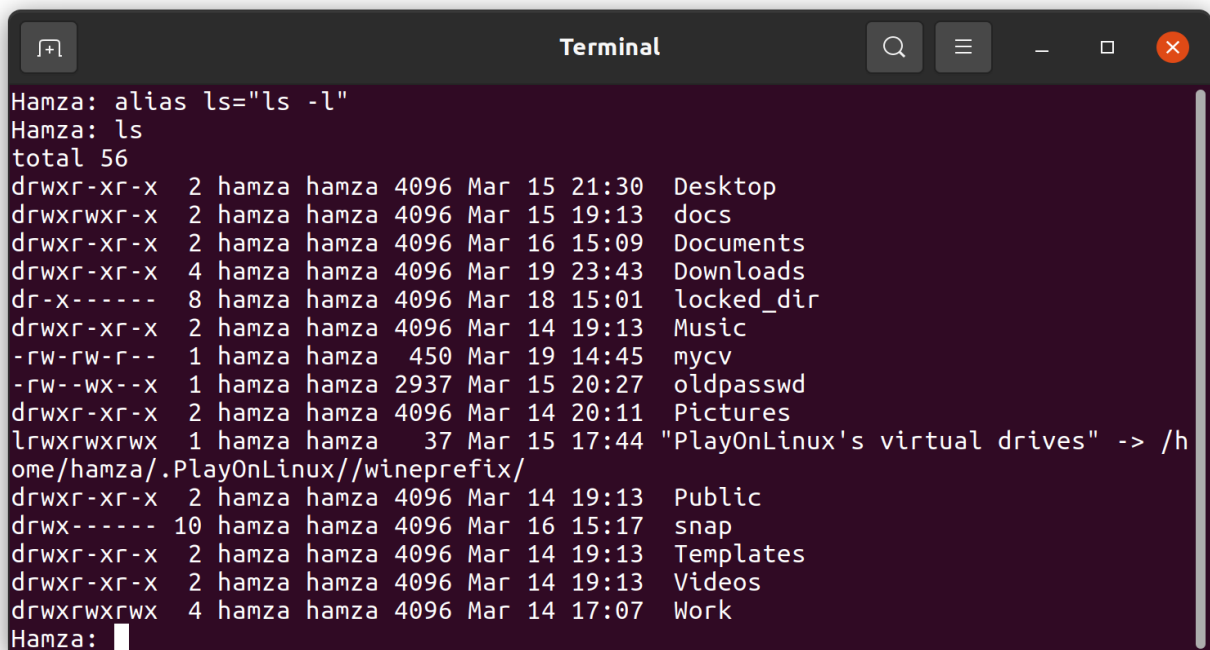


```
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 echo "Hello $USER, today's date is $(date +%A, %d-%m-%y)"
120 PS1='Hamza: '
121 PS2=': '
```



```
Terminal
Hello hamza, today's date is Sunday, 19-03-23
Hamza: echo \
: 
```

11. Create a Bash shell alias named `ls` for the “`ls -l`” command



```
Terminal
Hamza: alias ls="ls -l"
Hamza: ls
total 56
drwxr-xr-x  2 hamza hamza 4096 Mar 15 21:30 Desktop
drwxrwxr-x  2 hamza hamza 4096 Mar 15 19:13 docs
drwxr-xr-x  2 hamza hamza 4096 Mar 16 15:09 Documents
drwxr-xr-x  4 hamza hamza 4096 Mar 19 23:43 Downloads
dr-x----- 8 hamza hamza 4096 Mar 18 15:01 locked_dir
drwxr-xr-x  2 hamza hamza 4096 Mar 14 19:13 Music
-rw-rw-r--  1 hamza hamza  450 Mar 19 14:45 mycv
-rw--wx--x  1 hamza hamza 2937 Mar 15 20:27 oldpasswd
drwxr-xr-x  2 hamza hamza 4096 Mar 14 20:11 Pictures
lrwxrwxrwx  1 hamza hamza   37 Mar 15 17:44 "PlayOnLinux's virtual drives" -> /home/hamza/.PlayOnLinux//wineprefix/
drwxr-xr-x  2 hamza hamza 4096 Mar 14 19:13 Public
drwx----- 10 hamza hamza 4096 Mar 16 15:17 snap
drwxr-xr-x  2 hamza hamza 4096 Mar 14 19:13 Templates
drwxr-xr-x  2 hamza hamza 4096 Mar 14 19:13 Videos
drwxrwxrwx  4 hamza hamza 4096 Mar 14 17:07 Work
Hamza: 
```