System and Network Administration - Lab 3 - Command line and file manipulation

Jaffar Totanji - j.totanji@innopolis.university

Exercises:

```
kuro@kuro-VirtualBoxZorinOS:~$ pwd
/home/kuro
kuro@kuro-VirtualBoxZorinOS:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
kuro@kuro-VirtualBoxZorinOS:~$ mkdir mydocs
kuro@kuro-VirtualBoxZorinOS:~$ cd mydocs
kuro@kuro-VirtualBoxZorinOS:~/mydocs$ touch textproc
kuro@kuro-VirtualBoxZorinOS:~/mydocs$ ls -la
total 8
drwxrwxr-x  2 kuro kuro 4096 сен 13 15:51 .
drwxr-xr-x 15 kuro kuro 4096 сен 13 15:51 ..
-rw-rw-r--  1 kuro kuro    0 сен 13 15:51 textproc
kuro@kuro-VirtualBoxZorinOS:~/mydocs$ cd ..
kuro@kuro-VirtualBoxZorinOS:~$ cp -R mydocs/textproc ~
kuro@kuro-VirtualBoxZorinOS:~$ ls
Desktop    Downloads  mydocs    Public     textproc
Documents  Music      Pictures  Templates  Videos
kuro@kuro-VirtualBoxZorinOS:~$ echo "My name is $USER and my home directory is $HOME" > simple_echo
kuro@kuro-VirtualBoxZorinOS:~$ cat simple_echo
My name is kuro and my home directory is /home/kuro
kuro@kuro-VirtualBoxZorinOS:~$ echo "My salary is "$"100" >> simple_echo
kuro@kuro-VirtualBoxZorinOS:~$ cat simple_echo
My name is kuro and my home directory is /home/kuro
My salary is 100
kuro@kuro-VirtualBoxZorinOS:~$ cat simple_echo > new_echo
kuro@kuro-VirtualBoxZorinOS:~$ cat nofile
cat: nofile: No such file or directory
kuro@kuro-VirtualBoxZorinOS:~$ cat nofile 2> error_out
kuro@kuro-VirtualBoxZorinOS:~$ cat nofile > allout 2>&1
kuro@kuro-VirtualBoxZorinOS:~$ cat allout
cat: nofile: No such file or directory
kuro@kuro-VirtualBoxZorinOS:~$ cat << foobar
> Hello foobar
> foobar
Hello foobar
kuro@kuro-VirtualBoxZorinOS:~$ touch fragmented.txt
```

```
kuro@kuro-VirtualBoxZorinOS:~$ vim fragmented.txt
kuro@kuro-VirtualBoxZorinOS:~$ nl < fragmented.txt
     1  hi
     2  this
     3  is
     4  some
     5  fragmented
     6  text
kuro@kuro-VirtualBoxZorinOS:~$ nl fragmented.txt
     1  hi
     2  this
     3  is
     4  some
     5  fragmented
     6  text
kuro@kuro-VirtualBoxZorinOS:~$ sort << end
> hi
> this is some text
> that is about to be sorted
> very nice
> end
hi
that is about to be sorted
this is some text
very nice
kuro@kuro-VirtualBoxZorinOS:~$ sort << end > sorted
> hi
> this text is about to be sorted
> then stored in a file
> called
> sorted
> ironic
> end
kuro@kuro-VirtualBoxZorinOS:~$ cat sorted
called
hi
ironic
```

```
sorted
then stored in a file
this text is about to be sorted
```

```
kuro@kuro-VirtualBoxZorinOS:~$ sort << end > sorted
> hi
> this text is about to be sorted
> then stored in a file
> called
> sorted
> ironic
> end
kuro@kuro-VirtualBoxZorinOS:~$ cat sorted
called
hi
ironic
sorted
then stored in a file
this text is about to be sorted
kuro@kuro-VirtualBoxZorinOS:~$ sort << end | nl -ba > sorted_numbered
> hi
> this is some text
> that is about to be
> sroted
> and
> numbered
> end
kuro@kuro-VirtualBoxZorinOS:~$ cat sorted_numbered
     1  and
     2  hi
     3  numbered
     4  sroted
     5  that is about to be
     6  this is some text
kuro@kuro-VirtualBoxZorinOS:~$ echo -e "Next is the \nNew line"
Next is the
New line
kuro@kuro-VirtualBoxZorinOS:~$ cat /etc/passwd | grep $USER
kuro:x:1000:1000:kuro,,,:/home/kuro:/bin/bash
```

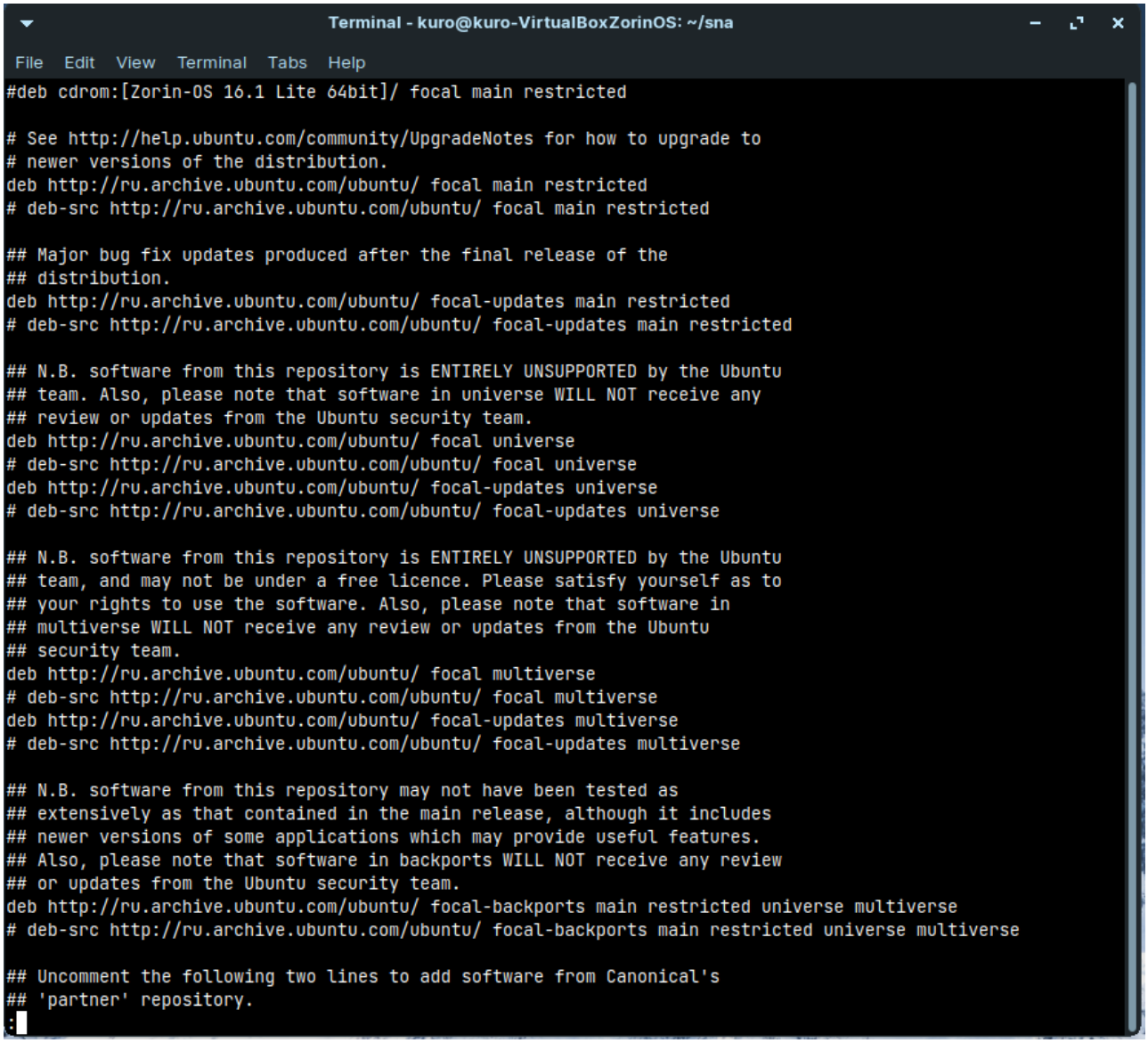
```
kuro@kuro-VirtualBoxZorinOS:~$ mkdir sna
kuro@kuro-VirtualBoxZorinOS:~$ cd sna/
kuro@kuro-VirtualBoxZorinOS:~/sna$ echo -e "for1\nfor2\nfor3\nfor4\nfor5\nfor6\nfor7\nfor8\nfor9\nfor10" > n
umbers
kuro@kuro-VirtualBoxZorinOS:~/sna$ tail -n5 numbers
for6
for7
for8
for9
for10
kuro@kuro-VirtualBoxZorinOS:~/sna$ head -n3 numbers
for1
for2
for3
kuro@kuro-VirtualBoxZorinOS:~/sna$ tac numbers
for10
for9
for8
for7
for6
for5
for4
for3
for2
for1
kuro@kuro-VirtualBoxZorinOS:~/sna$
```

Questions to answer:

1. The Pipe or `|` in linux is used to send the output of one command to another for further processing. In this command `cat /etc/apt/sources.list | less` specifically, the pipe passes the output

of `cat /etc/apt/sources.list` to `less` which then displays the output one page at a time, and is usually used to view large files.

```
kuro@kuro-VirtualBoxZorinOS:~/sna$ cat /etc/apt/sources.list | less
```



2. `man` is divided into 8 different sections. There are certain terms that have different pages in different sections (e.g. printf as a command appears in section 1, as a `stdlib` function appears in section 3). Section 5 of `man` contains information about `File formats and conventions e.g. /etc/passwd.`. To tell `man` explicitly to open a specific section for our search, we can simply use:

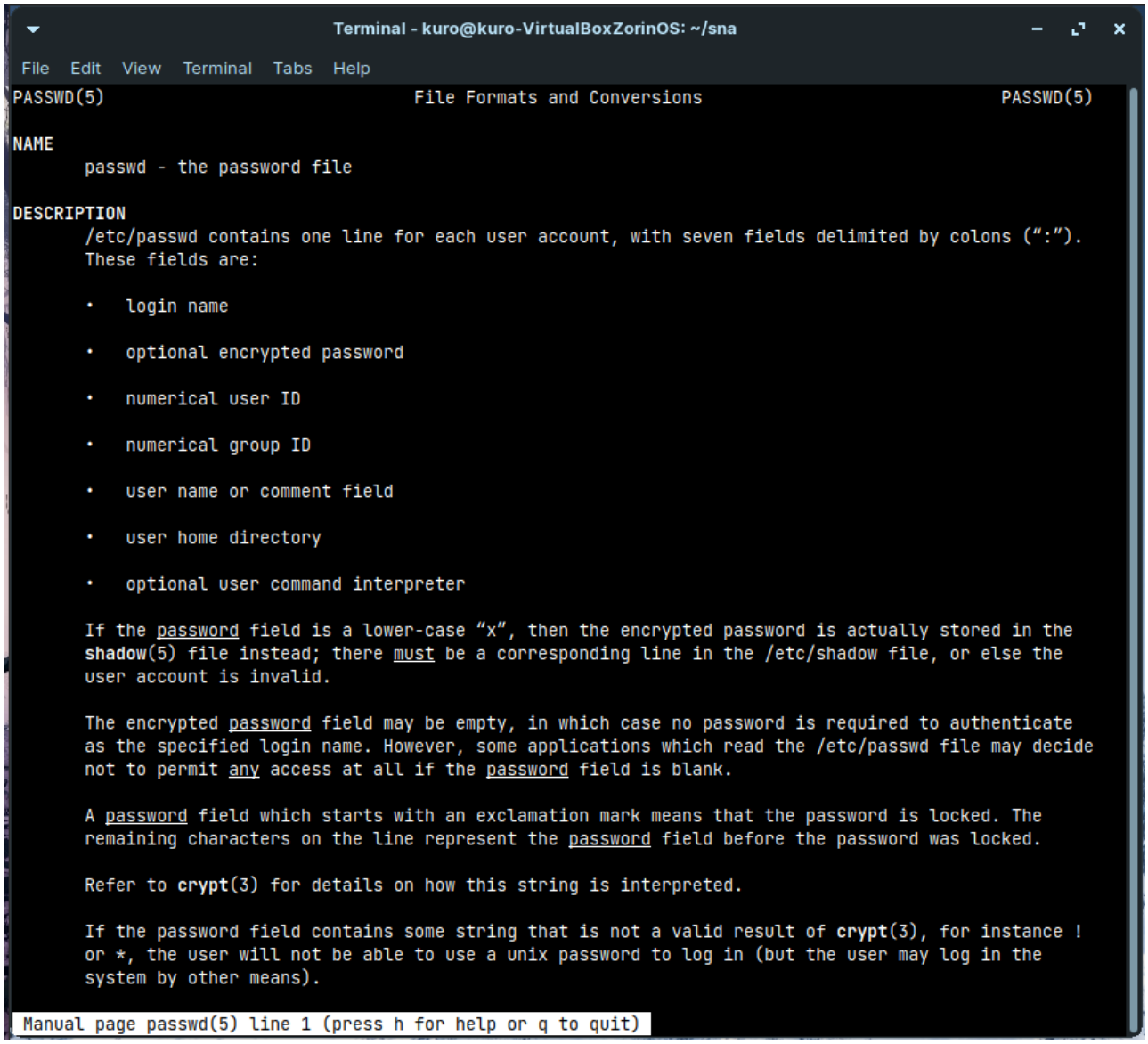
   ```
   $ man [SECTION-NUM] [COMMAND NAME]
   ```

   and for section 5 we can use:

   ```
   $ man 5 [COMMAND NAME]
   ```

   for example:

```
kuro@kuro-VirtualBoxZorinOS:~/sna$ man 5 passwd
```



3. `ls` command on my virtual machine is located at `/usr/bin/ls`. We can find that using:

```
whereis ls
```

which shows the location of the command followed by its `man` page path.

```
kuro@kuro-VirtualBoxZorinOS:~/sna$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

4. Some of the ways to rename the file are:

```
mv test_file.tot test_file.txt
```

```
kuro@kuro-VirtualBoxZorinOS:~/sna$ ls
test_file.tot
kuro@kuro-VirtualBoxZorinOS:~/sna$ mv test_file.tot test_file.txt
kuro@kuro-VirtualBoxZorinOS:~/sna$ ls
test_file.txt
kuro@kuro-VirtualBoxZorinOS:~/sna$
```

```
rename -v 's/.tot/.txt/' *.tot
```

```
kuro@kuro-VirtualBoxZorinOS:~/sna$ ls
test_file.tot
kuro@kuro-VirtualBoxZorinOS:~/sna$ rename -v 's/.tot/.txt/' *.tot
test_file.tot renamed as test_file.txt
kuro@kuro-VirtualBoxZorinOS:~/sna$ ls
test_file.txt
kuro@kuro-VirtualBoxZorinOS:~/sna$
```

5. First, I saved the given text to a file for readability of the required command, for that I used:

```
echo -e "The location of hundreds of crab pots\nLittle Red Riding
Hood\nThe location of hundreds of crab pots\nThe location of hundreds
of crab pots\nThe sound of thunder\nEight hours in a row\nAll
aboard\nEight hours in a row" > test.txt
```

```
kuro@hp-pavilion:~/test$ echo -e "The location of hundreds of crab pots\nLittle
Red Riding Hood\nThe location of hundreds of crab pots\nThe location of hundreds
 of crab pots\nThe sound of thunder\nEight hours in a row\nAll aboard\nEight hou
rs in a row" > test.txt
kuro@hp-pavilion:~/test$ cat test.txt
The location of hundreds of crab pots
Little Red Riding Hood
The location of hundreds of crab pots
The location of hundreds of crab pots
The sound of thunder
Eight hours in a row
All aboard
Eight hours in a row
kuro@hp-pavilion:~/test$
```

The command simply outputs the given text but redirects it to a file called `test.txt`.

Then, I used the following compound command:

```
cat test.txt | sort | uniq -u > unique.txt && whoami >> unique.txt
```

```
kuro@hp-pavilion:~/test$ cat test.txt | sort | uniq -u > unique.txt && whoami >>
 unique.txt
kuro@hp-pavilion:~/test$ cat unique.txt
All aboard
Little Red Riding Hood
The sound of thunder
kuro
kuro@hp-pavilion:~/test$
```

The commands simply do the following in order: - Output the contents of `test.txt` (the given string). - Pick up the forwarded output from the previous command and sort it. - Pick up the forwarded (now sorted) output of the previous command and filter out any lines that are not unique. - Redirect the output of the previous command (sorted unique lines) to a file called `unique.txt`. - Get the name of the currently logged in user and append it to the file `unique.txt`.

6. We can simply redirect `stdout` and `stderr` to a file using:

```
ping 127.0.0.1 &> test.txt
```

```
kuro@kuro-VirtualBoxZorinOS:~/sna$ ping 127.0.0.1 &> test.txt
^Ckuro@kuro-VirtualBoxZorinOS:~/sna$ cat test.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.068 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.077 ms

--- 127.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5099ms
rtt min/avg/max/mdev = 0.044/0.061/0.077/0.011 ms
kuro@kuro-VirtualBoxZorinOS:~/sna$
```

7. We can sort our input using `sort` and add numbers to the sorted lines regardless of blank lines using `nl -ba` and then save it to a file by redirecting `stdout`.

```
sort << end | nl -ba > sorted_input
```

```
nonroot@kuro-VirtualBoxZorinOS:~/testdir$ sort << end | nl -ba > sorted_input
> hi
> this is
> some input
>
> and a blank line
> i will now perform magic
> end
nonroot@kuro-VirtualBoxZorinOS:~/testdir$ cat sorted_input
     1
     2  and a blank line
     3  hi
     4  i will now perform magic
     5  some input
     6  this is
nonroot@kuro-VirtualBoxZorinOS:~/testdir$
```

8. A couple of ways to navigate between the 2 directories using `cd`:

```
nonroot@kuro-VirtualBoxZorinOS:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
nonroot@kuro-VirtualBoxZorinOS:~$ mkdir testdir
nonroot@kuro-VirtualBoxZorinOS:~$ cd /usr/share/
nonroot@kuro-VirtualBoxZorinOS:/usr/share$ cd /home/nonroot/testdir/
nonroot@kuro-VirtualBoxZorinOS:~/testdir$ cd /usr/share/
```

```
nonroot@kuro-VirtualBoxZorinOS:~/testdir$ cd /usr/share/
nonroot@kuro-VirtualBoxZorinOS:/usr/share$ cd ..
nonroot@kuro-VirtualBoxZorinOS:/usr$ cd ..
nonroot@kuro-VirtualBoxZorinOS:/$ ls
bin    dev   lib    libx32      mnt   root  snap        sys  var
boot   etc   lib32  lost+found  opt   run   srv         tmp
cdrom  home  lib64  media       proc  sbin  swapfile    usr
nonroot@kuro-VirtualBoxZorinOS:/$ cd home/
nonroot@kuro-VirtualBoxZorinOS:/home$ cd nonroot/
nonroot@kuro-VirtualBoxZorinOS:~$ cd testdir/
nonroot@kuro-VirtualBoxZorinOS:~/testdir$
```

```
nonroot@kuro-VirtualBoxZorinOS:~/testdir$ cd /usr/share/
nonroot@kuro-VirtualBoxZorinOS:/usr/share$ cd ~
nonroot@kuro-VirtualBoxZorinOS:~$ cd testdir/
nonroot@kuro-VirtualBoxZorinOS:~/testdir$
```

```
nonroot@kuro-VirtualBoxZorinOS:/usr/share$ cd /home/$(whoami)/testdir
nonroot@kuro-VirtualBoxZorinOS:~/testdir$
```

9. The pipe is as follows:

```
cat /etc/passwd | rev | cut -d":" -f1 | rev | sort | uniq
```

Basically outputting the contents of `/etc/passwd` then reversing them with `rev` and getting the first column (last column originally) with `cut -d":" -f1` then reversing them back, then finally sorting the removing duplicates. Here is the command in action:

```
kuro@kuro-VirtualBoxZorinOS:~/Desktop$ cat /etc/passwd | rev | cut -d":" -f1 | r
ev | sort | uniq
/bin/bash
/bin/false
/bin/sync
/usr/sbin/nologin
kuro@kuro-VirtualBoxZorinOS:~/Desktop$
```

10. We can use:

```
man -wK malloc
```

Which uses 2 flags that do the following in order:

- Print the locations of the manual pages instead of showing the pages themselves.
- Search for the given text in all manual pages.

We can see the result of the previous command:

```
nonroot@kuro-VirtualBoxZorinOS:~/testdir$ man -wK malloc
/usr/share/man/man1/python3.8.1.gz
/usr/share/man/man1/python3.8.1.gz
/usr/share/man/man1/ltrace.1.gz
/usr/share/man/man1/memusage.1.gz
/usr/share/man/man1/mtrace.1.gz
/usr/share/man/man1/top.1.gz
/usr/share/man/man1/xterm.1.gz
/usr/share/man/man8/ld.so.8.gz
/usr/share/man/man8/ld.so.8.gz
/usr/share/man/man8/ld.so.8.gz
/usr/share/man/man3/libmaxminddb.3.gz
/usr/share/man/man3/libmaxminddb.3.gz
/usr/share/man/man3/readline.3readline.gz
/usr/share/man/man3/pcrepattern.3.gz
/usr/share/man/man5/ltrace.conf.5.gz
/usr/share/man/man5/proc.5.gz
/usr/share/man/man5/proc.5.gz
/usr/share/man/man5/slabinfo.5.gz
/usr/share/man/man7/aio.7.gz
/usr/share/man/man7/attributes.7.gz
/usr/share/man/man7/bpf-helpers.7.gz
/usr/share/man/man7/environ.7.gz
/usr/share/man/man7/unix.7.gz
nonroot@kuro-VirtualBoxZorinOS:~/testdir$
```

End of Exercises