```
johnson
victoria
$ _
```

Note The options of the sort command can be combined together to get the required form of output. Some of the combinations are as follows:

-nr or -rn	(combination of the -n and -r option)
-uf or -fu	(combination of the -u and -f option)
+4n -5	(combination of the +pos -pos and -n option)
+3nr -2 or +3rn –2	(combination of +pos -pos, -n and -r options)

The sort command like all other LINUX commands can be used with pipes.

The grep Filter

The grep command is used to search for a particular pattern from a file or from the Standard Input and display those lines on the Standard Output. "grep" stands for "global search for regular expression".

The various options available with the grep command are:

-v	Displays only those lines that do not match the specified pattern.
-c	Displays only the count of the lines that match the specified pattern.
-n	Displays those lines the match the specified pattern along with the line number at the beginning of the line.
-i	Displays those lines that match the specified pattern ignoring the case distinction.

There are various ways to specify the regular expressions or the patterns to be searched. We shall try understanding them using examples.

Example 4.13

Let us assume that we have a file called **data** whose contents are as follows:

```
$ cat data
india           6890    asia
china           8765    Asia
france          3243    europe
nigeria         3212    Africa
argentina       1234    south america
mexico          4563    north america
```

```
$ _
```

To extract the lines that have the text "asia", the command is:

```
$ grep "asia" data
india     6890      asia
$ _
```

Example 4.14

If we want the above example to ignore the case distinction then the command is:

```
$ grep -i  "asia"  data
india   6890   asia
china   8765   Asia
$ _
```

Here we shall recall the usage of "[ ]" type of Wild card patterns. It is also used with grep in the same way.

Example 4.15

Consider a file called **new**, whose contents are as follows:

```
$ cat new
this is a new file.

there are two types of files
newa, newb and newc are
the first type. The second
is new1, new2, new3.
$ _

$ grep "new[a-c]" new
newa, newb and newc are
$ _
```

To extract the lines which end with the character "e", the command to be issued is:

Example 4.16

```
$ grep "e$" new
this is a new file.
newa, newb and newc are
$ _
```

If we want to extract the lines that begin with the letter "t" the command is:

Example 4.17

```
$ grep "^t" new
this is a new file.
there are two types of files,
the first type.  The second
$ _
```

The egrep Command

The egrep command is an extension of the grep command. "egrep" means extended global search for regular expression. This offers a few more additional features than grep.

Multiple patterns can be searched in a file by issuing a single command. These multiple patterns should be separated by a pipe (|) symbol.

Example 4.18

Let us consider the file **data** of example 4.13. If we want to search for the lines that contain the words "india", "nigeria", "argentina", then the command is:

```
$ egrep "india|nigeria|argentina"  data

india      6890       asia
nigeria    3212       africa
argentina 1234        south america
$ _
```

We can also make egrep to look out for patterns from a different file. For instance consider a file called **countries** whose contents are as follows.

```
$ cat countries
france
india
$ _
```

Then, the command to look for data in both the files is:

```
$ egrep -f  countries data
france     3243       europe
india      6890       asia
$ _
```

The egrep command allows us to enclose the required pattern within parenthesis. For example, if we want to extract the lines ending with "ica" then the command is:

Example 4.19

```
$ egrep "(ica)$" data
argentina 1234        south america
mexico     4563       north america
nigeria    3212       africa
$ _
```

The fgrep Command

fgrep stands for "fixed grep". This command is used to extract only fixed strings without use of any regular expression.

fgrep when used with -x option is used to extract those lines which match the string exact

Example 4.20

Let us consider a file called **old** whose contents are as follows:

```
$ cat old
this is a test
to extract the exact string.

this is only possible
by this command.
$ _
```

Suppose we specify the string to be searched as "this is", then

```
$ fgrep "this is" old
this is a test
this is only possible
$ _
```

B. Points to Ponder

State True or False

1. The sort -n option will arrange the input according to their numerical value.
2. The +pos -pos option is used to sort a particular field.
3. The egrep command extracts the fixed string and displays it.
4. The grep -v option will display those lines that match the specified pattern.
5. The egrep command can take patterns from external files also.

Fill in the blanks

1. The __________ option will help the user to dump the sorted contents to a file.
2. The sort -b option helps in ignoring the __________.
3. The command to extract lines that begin with a pattern is ________.
4. The command to extract lines that end with a pattern is ________.
5. If we want to extract more than one pattern then the command is ________.

Usage of Other Filters

We shall now see the usage of certain other filters that help us to work better in the LINUX environment.

The uniq Filter

The uniq filter compares the adjacent lines in the sorted input file. When used with different options, the uniq filter displays the single occurrences and multiple occurrences. Since it is a filter, it normally takes its input from the Standard Input and displays the output to the Standard Output.

Example 4.21

Let us consider the following file called **mast**, which is sorted.

```
$ cat mast
b001:kane and abel:j archer
b001:kane and abel:j archer
b001:kane and abel:j archer
b003:the naked face:s sheldon
b003:the naked face:s sheldon
b005:where eagles dare:a maclean
b006:the ring:d steele
b006:the ring:d steele
b008:the almighty: I wallace
$ _
```

```
$ uniq mast
b001:kane and abel:j archer
b003:the naked face:s sheldon
b005:where eagles dare:a maclean
b006:the ring:d steele
b008:the almighty: I wallace
$ _
```

The different options of the uniq command are:

-d	Displays only the lines that are duplicated in the input file.
-u	Displays only the lines with single occurrences.
-c	Precedes each line displayed by the number of times it occurs.

The more Filter

The more filter displays the output of a command on the screen page by page. The next screen can be viewed by pressing space bar.

```
$ who | more
```

The above command displays the list of people logged in one screen at a time.