

```
3  bsd      1000
4  linux     1000
1  mac       2000
2  winxp     4000
```



Always be careful about the `-n` option for numeric sort. The `sort` command treats alphabetical sort and numeric sort differently. Hence, in order to specify numeric sort the `-n` option should be provided.

Usually, by default, keys are columns in the text file. Columns are separated by space characters. But, in certain circumstances, we may need to specify keys as a group of characters in the given character number range (for example, `key1 = character4-character8`). In such cases where keys are to be specified explicitly as a range of characters, we can specify the key as ranges with the character position at key starts and key ends as follows:

```
$ cat data.txt
1010hellothis
2189ababbba
7464dfddfdfd
$ sort -nk 2,3 data.txt
```

The highlighted characters are to be used as numeric keys. To extract, use their positions in the lines as the key format (in the previous example, they're 2 and 3).

To use the first character as the key, use:

```
$ sort -nk 1,1 data.txt
```

To make the sort's output `xargs` compatible with the `\0` terminator, use the following command:

```
$ sort -z data.txt | xargs -0
#Zero terminator is used to make safe use with xargs
```

Sometimes, the text may contain unnecessary extraneous characters such as spaces. To sort them in dictionary order, by ignoring punctuations and folds, use:

```
$ sort -bd unsorted.txt
```

The `-b` option is used to ignore leading blank lines from the file and the `-d` option is used to specify sort in the dictionary order.

## uniq

`uniq` is a command used to find out the unique lines from the given input (`stdin` or from a filename as command argument) by eliminating the duplicates. It can also be used to find out the duplicate lines from the input.

`uniq` can be applied only for sorted data input. Hence, `uniq` is to be used always along with the `sort` command using pipe or using a sorted file as input.

Produce the unique lines (all lines in the input are printed and even the duplicate lines are printed only once) from the given input data as follows:

```
$ cat sorted.txt
bash
foss
hack
hack
```

```
$ uniq sorted.txt
bash
foss
hack
```

Or:

```
$ sort unsorted.txt | uniq
```

Display only unique lines (the lines which are not repeated or duplicated in the input file) as follows:

```
$ uniq -u sorted.txt
bash
foss
```

Or:

```
$ sort unsorted.txt | uniq -u
```

To count how many times each of the lines appears in the file, use the following command:

```
$ sort unsorted.txt | uniq -c
  1 bash
  1 foss
  2 hack
```