

Bash Assignment Hints

to get the second field of each line in file "triples"

```
awk '{print $2}' triples
```

to get the second field of all files that start with the string "tri"

```
awk '{print $2}' tri*
```

to save the result to the file "col2"

```
awk '{print $2}' tri* > col2
```

to append the result to the file "col2"

```
awk '{print $2}' tri* >> col2
```

to sort the file "carmen"

```
sort carmen
```

to sort the file "carmen" by the second field

```
sort -k 2 carmen
```

to find the five largest numbers in the third field of "carmen"

```
sort -k 3 -nr carmen | head -n 3
```

to substitute all dashes with spaces

```
cat carmen.txt | sed 's/-/ /g' > carmen-no-dashes.txt
```

to find all lines containing the string "bog" in file "large"

```
grep bog large
```

to find all lines containing the string "bog" followed by a space

```
grep "bog " large
```

to save the result to "boglines"

```
grep bog large > boglines
```

to save a sorted list of the third field of all lines in large that contain the string bog

```
grep bog large | awk '{print $3}' | sort > boglines
```

find all lines with the word "bog" at the end of the line

```
grep "bog$" large
```

to find all lines in file that contain a digit

```
grep "[0-9]" file
```

to find all lines in file that contain a three-digit string

```
grep "[0-9][0-9][0-9]" file
```

to find all lines in file not containing a three-digit string

```
grep -v "[0-9][0-9][0-9]" file
```

to learn more about grep

```
man grep
```

to learn more about join

```
man join
```

to join two sorted files (for each pair of lines with the same first field, output a line containing all fields of both lines)

```
join file1 file2
```

For example if file1 is

```
carlos smith  
josh hernandez  
zaid Suzuki
```

and file 2 is

```
carlos m 23  
elfie f 19  
josh m 21
```

then the output will be

```
carlos smith m 23  
josh hernandez m 21
```

Cygwin Notes

If creating a script and running it under cygwin (on a Windows machine):

- 1) the first two lines of your bash script should be:

```
#!/usr/bin/bash  
(set -o igncr) 2>/dev/null && set -o igncr; # comment is needed
```

- 2) If you get "command not found" errors when you try to run your script, e.g. myscript.sh, this may be due to different encodings for line endings, and may be fixable with `dos2unix myscript.sh`