

### Assignment 6 – sed and awk

**Commands:** sed [option] [script] [input file]  
awk [options] 'selection criteria/pattern' {action} input file > output file

#### **Exercise 1:**

**1. Launch the terminal.** - A terminal window was opened.

**2. Create and save the following file.** Note that the fields are separated by one or more spaces (randomly). The first field is the first name, the second field is the last name, and the third field is the age.

John Adams 55

George Bull 77

Anne Blue 99

Janet Blue 67

Ben Benjamin 78

Ted White 32

kri@kri-ubuntu:~\$ cat > ex1.txt

John Adams 55

George Bull 77

Anne Blue 99

Janet Blue 67

Ben Benjamin 78

Ted White 32

kri@kri-ubuntu:~\$ cat ex1.txt

John Adams 55

George Bull 77

Anne Blue 99

Janet Blue 67

Ben Benjamin 78

Ted White 32

kri@kri-ubuntu:~\$ ls

ex1.txt

**Description:** using cat > a new file called ex1.txt is created and the above input is given.

**3. Use a one-line sed command to reorganize the file using the comma/tab pattern shown in the following file.** Note that the last name is before the first name, and there is only one space between the names and a space before the numbers. Use the same name for the new file.

Adams, John 55

Bull, George 77

Blue, Anne 99

Blue, Janet 67

Benjamin, Ben 78

White, Ted 32

kri@kri-ubuntu:~\$ sed -i "s/^\(.\*\) \(.\*\) \(.\*\)/2, \1 \3/" ex1.txt

kri@kri-ubuntu:~\$ cat ex1.txt

Adams, John 55

Bull, George 77  
Blue, Anne 99  
Blue, Janet 67  
Benjamin, Ben 78  
White, Ted 32

**Description:** sed -i is inplace editing using sed. The regular expression saves the first 3 fields in the first three buffers separated by space. The buffers are then reordered as \2, \1 \3 using the substitute option if sed.

**4. Sort the file first according to the last name and then according to the age. Watch out for the comma after the last name. Use the same name for the new file.**

```
kri@kri-ubuntu:~$ sort -k1,1 -k3 ex1.txt > output.txt; cp output.txt ex1.txt; rm output.txt  
kri@kri-ubuntu:~$ cat ex1.txt
```

Adams, John 55  
Benjamin, Ben 78  
Blue, Janet 67  
Blue, Anne 99  
Bull, George 77  
White, Ted 32

**Description:** sort the first field using key 1 and using comma as delimiter and then sort using key field 3 and redirect contents back to the same file.

**5. Use a sed script and a sed command to put a set of five asterisks at the beginning and end of each line that contains the pattern Blue.**

```
kri@kri-ubuntu:~$ sed 'Blue/ s/^.*$/*****&*****/' ex1.txt
```

Adams, John 55  
Benjamin, Ben 78  
\*\*\*\*\*Blue, Janet 67\*\*\*\*\*  
\*\*\*\*\*Blue, Anne 99\*\*\*\*\*  
Bull, George 77  
White, Ted 32

**Description:** The sed command searches for the pattern Blue. Then the substitute command is used to copy all the contents of the matched line to the pattern buffer represented by \$. Then the \$ is added in between 10 asterisks as the replacement pattern.

**6. Use the cat command to insert a line number at the beginning of each line in a file.**

```
kri@kri-ubuntu:~$ cat -n ex1.txt > output.txt; cp output.txt ex1.txt; rm output.txt  
kri@kri-ubuntu:~$ cat ex1.txt
```

```
1 Adams, John 55  
2 Benjamin, Ben 78  
3 Blue, Janet 67  
4 Blue, Anne 99  
5 Bull, George 77  
6 White, Ted 32
```

**Description:** cat -n appends line number to the beginning of each line. Redirection to save output.

7. Write a sed script and a sed command to split the file into three files. The first file, called f1, contains lines 2 and 3. The second file, called f2, contains lines 4 and 5. The third file, called f3, contains lines 1 and 6.

```
kri@kri-ubuntu:~$ sed -nf f1.sed ex1.txt > f1; sed -nf f2.sed ex1.txt > f2; sed -nf f3.sed ex1.txt > f3
```

```
kri@kri-ubuntu:~$ cat f1
```

```
2 Benjamin, Ben 78
3 Blue, Janet 67
```

```
kri@kri-ubuntu:~$ cat f2
```

```
4 Blue, Anne 99
5 Bull, George 77
```

```
kri@kri-ubuntu:~$ cat f3
```

```
1 Adams, John 55
2 Benjamin, Ben 78
3 Blue, Janet 67
4 Blue, Anne 99
5 Bull, George 77
6 White, Ted 32
```

**Description:** the sed script for each operation contains the lines and the print option. While calling sed -f denotes sed script and -n suppresses the printing of lines that do not match. The output of sed is redirected to the new files.

8. Print all of the files created in this session and verify the output.

```
kri@kri-ubuntu:~$ ls
```

```
ex1.txt f1 f1.sed f2 f2.sed f3 f3.sed
```

**Description:** ls is used to list the contents of the directory.

9. Quit the terminal.

```
kri@kri-ubuntu:~$ exit
```

(The terminal window is closed.)

**Description:** The exit command can be used to quit the terminal.

---

## Exercise 2:

1. Launch the terminal. - A terminal window was opened.

2. Create the following file and call it a7-e4-f1. Each line in the file is an absolute pathname of a file.

```
bin/date
```

```
bin/programs/cal
```

```
usr/bin/date
```

```
usr/report/file1
```

```
usr/report/letters/lett1
```

```
spool/mails
```

```
kri@kri-ubuntu:~$ cat > a7-e4-f1
```

```
bin/date
bin/programs/cal
usr/bin/date
usr/report/file1
usr/report/letters/lett1
spool/emails
kri@kri-ubuntu:~$ cat a7-e4-f1
bin/date
bin/programs/cal
usr/bin/date
usr/report/file1
usr/report/letters/lett1
spool/emails
```

3. Write a sed script (a7-e4-f2) and a sed command to extract the lowest level directory and the name of the file from the path (separated by spaces) and store it in a file called a7-e4-f3. The file should look like the following (directory then file):

```
/bin date
/bin/programs cal
/usr/bin date
/usr/report file1
/usr/report/letters lett1
/spool emails
kri@kri-ubuntu:~$ cat > a7-e4-f2.sed
s/(.*)\\V\\1 /
kri@kri-ubuntu:~$ sed -f a7-e4-f2.sed a7-e4-f1 > a7-e4-f3
kri@kri-ubuntu:~$ cat a7-e4-f3
/bin date
/bin/programs cal
/usr/bin date
/usr/report file1
/usr/report/letters lett1
/spool emails
```

#### 4. Quit the terminal

```
kri@kri-ubuntu:~$ exit
```

(The terminal window is closed.)

**Description:** The exit command can be used to quit the terminal.

---

### Exercise 3:

1. **Launch the terminal.** - A terminal window was opened.

2. **Create the following file and call it a7-e6-f1.** The file is a C program that multiplies two numbers. It contains some comments which begin with the two-character token `/*` and end with the two-character token `*/`. In this program, comments can be on one line or can span more than a line

**/\* This program reads two integer numbers from the keyboard and prints their product.**

**Written by:**

**Date:**

**\*/**

**/\* Statements \*/**

```
scanf ("%d", &number1);  
scanf ("%d", &number2);  
result = number1 * number2;  
printf ("%d", result); return 0;  
} /* main */
```

**kri@kri-ubuntu:~\$ cat > a7-e6-f1**

**/\* and end with the two-character token \*/. In this program, comments can be on one line or can span more than a line**

**/\* This program reads two integer numbers from the keyboard and prints their product.**

**Written by:**

**Date:**

**\*/**

**/\* Statements \*/**

```
scanf ("%d", &number1);  
scanf ("%d", &number2);  
result = number1 * number2;  
printf ("%d", result); return 0;  
} /* main */
```

**kri@kri-ubuntu:~\$ cat a7-e6-f1**

**/\* and end with the two-character token \*/. In this program, comments can be on one line or can span more than a line**

**/\* This program reads two integer numbers from the keyboard and prints their product.**

**Written by:**

**Date:**

**\*/**

**/\* Statements \*/**

```
scanf ("%d", &number1);  
scanf ("%d", &number2);  
result = number1 * number2;  
printf ("%d", result); return 0;  
} /* main */
```

3. Write a sed script a7-e6-f2.sed and a sed command to delete the comments from the file.  
Call the new file a7-e6-f3.

```
kri@kri-ubuntu:~$  
//
```

4. Quit the terminal

```
kri@kri-ubuntu:~$ exit
```

(The terminal window is closed.)

**Description:** The exit command can be used to quit the terminal.

---

#### Exercise 4:

1. Launch the terminal. - A terminal window was opened.

2. Create the file named workers.txt with the following

First Name	Last Name	Rate	Hours
------------	-----------	------	-------

George	White	18.00	23
--------	-------	-------	----

Mark	Red	18.10	20
------	-----	-------	----

Mary	Blue	10.89	25
------	------	-------	----

Dan	Black	12.00	0
-----	-------	-------	---

Susan	Green	18.00	40
-------	-------	-------	----

Nora	Brown	17.20	46
------	-------	-------	----

Bruce	Purple	12.20	52
-------	--------	-------	----

John	Gray	11.00	39
------	------	-------	----

Bob	Gold	15.00	45
-----	------	-------	----

Steve	Silver	14.67	25
-------	--------	-------	----

```
kri@kri-ubuntu:~$ ccat > workers.txt
```

George	White	18.00	23
--------	-------	-------	----

Mark	Red	18.10	20
------	-----	-------	----

Mary	Blue	10.89	25
------	------	-------	----

Dan	Black	12.00	0
-----	-------	-------	---

Susan	Green	18.00	40
-------	-------	-------	----

Nora	Brown	17.20	46
------	-------	-------	----

Bruce	Purple	12.20	52
-------	--------	-------	----

John	Gray	11.00	39
------	------	-------	----

Bob	Gold	15.00	45
-----	------	-------	----

Steve	Silver	14.67	25
-------	--------	-------	----

```
kri@kri-ubuntu:~$ cat workers.txt
```

George	White	18.00	23
--------	-------	-------	----

Mark	Red	18.10	20
------	-----	-------	----

Mary	Blue	10.89	25
------	------	-------	----

Dan	Black	12.00	0
-----	-------	-------	---

Susan	Green	18.00	40
-------	-------	-------	----

Nora	Brown	17.20	46
------	-------	-------	----

Bruce	Purple	12.20	52
-------	--------	-------	----

John	Gray	11.00	39
------	------	-------	----

Bob	Gold	15.00	45
-----	------	-------	----

Steve Silver 14.67 25

a. Write awk command to print the first and last name who did not work in the last week.

```
kri@kri-ubuntu:~$ awk '$4==0 {print $1, $2}' workers.txt  
Dan Black
```

b. Write awk command to print the record of the employee whose rate is \$15 or more

```
kri@kri-ubuntu:~$ awk '$3>=15 {print $0}' workers.txt  
George White 18.00 23  
Mark Red 18.10 20  
Susan Green 18.00 40  
Nora Brown 17.20 46  
Bob Gold 15.00 45
```

c. Write awk command to print the record whose first name is Mary.

```
kri@kri-ubuntu:~$ awk '$1=="Mary" {print $0}' workers.txt  
Mary Blue 10.89 25
```

d. Write awk command to print the record of the employee whose rate between \$1 and \$18

```
kri@kri-ubuntu:~$ awk '$3>=1 && $3<=18 {print $0}' workers.txt  
George White 18.00 23  
Mary Blue 10.89 25  
Dan Black 12.00 0  
Susan Green 18.00 40  
Nora Brown 17.20 46  
Bruce Purple 12.20 52  
John Gray 11.00 39  
Bob Gold 15.00 45  
Steve Silver 14.67 25
```

3. Create a file sales.txt with the following contents.

Month	Sales
January	20
February	30
March	43
February	34
January	12
June	89
May	97
June	60
July	23
August	13
August	45
October	56
October	45
November	34

```
kri@kri-ubuntu:~$ cat > sales.txt
```

Month	Sales
January	20
February	30
March	43
February	34
January	12
June	89
May	97
June	60
July	23
August	13
August	45
October	56
October	45
November	34

```
kri@kri-ubuntu:~$ cat sales.txt
```

Month	Sales
January	20
February	30
March	43
February	34
January	12
June	89
May	97
June	60
July	23
August	13
August	45
October	56
October	45
November	34

a. Write awk command to find total sales

```
kri@kri-ubuntu:~$ awk '{sum+=$2;} END{print sum;}' sales.txt  
601
```

b. Write awk script to find the total sales in every month.

```
kri@kri-ubuntu:~$ awk '{a[$1] += $2} END{for (i in a){print i, a[i]}}' sales.txt  
June 149  
October 101  
March 43  
August 58  
July 23  
May 97  
February 64  
January 32  
November 34
```



**c. Write awk script to find the months with no sales**

```
kri@kri-ubuntu:~$ awk '{a[$1] += $2} END{for (i in a){if(a[i]==0){print i, a[i]]}}' sales.txt  
kri@kri-ubuntu:~$
```

**4. Quit the terminal**

```
kri@kri-ubuntu:~$ exit
```

(The terminal window is closed.)

**Description:** The exit command can be used to quit the terminal.

---