AWK command in UNIX

AWK command is a programming language that is executed byAWK interpreter.
Syntax for using AWK command is:

AWK option '/pattern/ {action}' file _name

option is optional. AWK command must have either pattern or action or both.
If pattern is not specified, it will be entire line.
If action is not specified, by default it is print.

Print all records of file Bank.lst

root@MUM084:~/Desktop# awk '{print}' bank.lst

```
101    ADITYA     0       14/11/2000  CURRENT
102    Anil   10000 20/05/2011  saving
103    Naman      0       20/08/2009  current
104    Ram    10000 15/08/2010  saving
105    Jyotsna        5000  16/06/2012  saving
106    Mukesh     14000 20/12/2009  Current
107    Vishal     14500 30/11/2011  saving
108    Chirag     0       15/12/2012  Current
109    Arya   16000 14/12/2010  Current
110    Priya  130    16/11/2009  Saving
201    Bina   3000   11/03/2010  saving
202    Diya   4000   13/04/2018  Saving
203    Gargi  2000   21/01/2015  current
Hina   204    30000 14/02/2014  saving
Kalpana        205    4000   8/9/2007      Current
301    Nikhil     7777   8/9/1999      saving
```

Print first 3 fields (acc no., name and balance) from bank.lst (field 1 is referred by $1 and so on.)

root@MUM084:~/Desktop# awk '{print $1 $2 $3}' bank.lst

```
101ADITYA0
102Anil10000
103Naman0
104Ram10000
105Jyotsna5000
106Mukesh14000
107Vishal14500
108Chirag0
```

```
109Arya16000
110Priya130
201Bina3000
202Diya4000
203Gargi2000
Hina20430000
Kalpana2054000
301Nikhil7777
```

Sepearate fields by tabs.

```
root@MUM084:~/Desktop# awk '{print $1 "\t" $2 "\t" $3}' bank.lst
101    ADITYA     0
102    Anil   10000
103    Naman      0
104    Ram    10000
105    Jyotsna        5000
106    Mukesh     14000
107    Vishal     14500
108    Chirag     0
109    Arya   16000
110    Priya  130
201    Bina   3000
202    Diya   4000
203    Gargi 2000
Hina   204    30000
Kalpana        205    4000
301    Nikhil     7777
```

Print only those records having 'current' account.

```
root@MUM084:~/Desktop# awk '/current/ {print}' bank.lst
103    Naman      0       20/08/2009  current
203    Gargi 2000  21/01/2015  current
```

OR

```
root@MUM084:~/Desktop# awk '/current/' bank.lst
103    Naman      0       20/08/2009  current
203    Gargi 2000  21/01/2015  current
```

OR

```
root@MUM084:~/Desktop# awk /current/ bank.lst
103    Naman      0       20/08/2009  current
203    Gargi 2000  21/01/2015  current
```

Format specifiers for the field can be specified as below:

```
bank.lst^Croot@MUM084:~/Desktop# awk '{printf "%3d \t %-15s \t %7d \n", $1, $2, $3}' bank.lst

101     ADITYA                0
102     Anil                  10000
103     Naman                 0
104     Ram                   10000
105     Jyotsna               5000
106     Mukesh                14000
107     Vishal                14500
108     Chirag                0
109     Arya                  16000
110     Priya                 130
201     Bina                  3000
202     Diya                  4000
203     Gargi                 2000
204     Hina                  30000
205     Kalpana               4000
301     Nikhil                7777
```

Printing records having 'current' account. $0 means entire line.

```
root@MUM084:~/Desktop# awk '/current/ {print $0}' bank.lst

103    Naman        0       20/08/2009  current
203    Gargi 2000  21/01/2015  current
```

Printing individual fields of file.

```
root@MUM084:~/Desktop# awk '/current/ {print $1}' bank.lst

103
203

root@MUM084:~/Desktop# awk '/current/ {print $2}' bank.lst
Naman
Gargi
```

Print records having balance less than 5000. (here $3 represents the 3rd field balance)

```
root@MUM084:~/Desktop# awk '$3<5000' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
103    Naman      0       20/08/2009  current
```

```
108    Chirag        0       15/12/2012  Current
110    Priya  130    16/11/2009  Saving
201    Bina   3000   11/03/2010  saving
202    Diya   4000   13/04/2018  Saving
203    Gargi 2000   21/01/2015  current
205    Kalpana       4000  8/9/2007      Current
```

'OR'ing two conditions.
Print records having balance less than 5000 or more than 10000

```
root@MUM084:~/Desktop# awk '$3<5000 || $3>10000' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
103    Naman      0       20/08/2009  current
106    Mukesh     1400020/12/2009  Current
107    Vishal     1450030/11/2011  saving
108    Chirag     0       15/12/2012  Current
109    Arya   1600014/12/2010  Current
110    Priya  130    16/11/2009  Saving
201    Bina   3000   11/03/2010  saving
202    Diya   4000   13/04/2018  Saving
203    Gargi 2000   21/01/2015  current
204    Hina   3000014/02/2014  saving
205    Kalpana       4000  8/9/2007      Current
```

Print records having balance less than 8000 and more than 3000

```
root@MUM084:~/Desktop# awk '$3>3000&&$3<8000 {print $1, $2, $3}' bank.lst
105 Jyotsna 5000
202 Diya 4000
205 Kalpana 4000
301 Nikhil 7777
```

Print all records whose account type is current.

```
root@MUM084:~/Desktop# awk '$5 == "current"' bank.lst
103    Naman      0       20/08/2009  current
203    Gargi 2000   21/01/2015  current
```

Print all records whose account type is Current.

```
root@MUM084:~/Desktop# awk '$5 == "Current"' bank.lst
106    Mukesh     1400020/12/2009  Current
108    Chirag     0       15/12/2012  Current
109    Arya   1600014/12/2010  Current
205    Kalpana       4000  8/9/2007      Current
```

Print all records whose account type is not Current.

```
root@MUM084:~/Desktop# awk '$5 != "Current"' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
102    Anil   10000020/05/2011  saving
103    Naman      0       20/08/2009  current
104    Ram    10000015/08/2010  saving
105    Jyotsna        5000  16/06/2012  saving
107    Vishal         14500030/11/2011  saving
110    Priya  130     16/11/2009  Saving
201    Bina   3000    11/03/2010  saving
202    Diya   4000    13/04/2018  Saving
203    Gargi  2000    21/01/2015  current
204    Hina   30000   14/02/2014  saving
301    Nikhil         7777  8/9/1999      saving
```

Print records whose acc. type is 'current"

```
root@MUM084:~/Desktop# awk '$5 ~/current/' bank.lst
103    Naman      0       20/08/2009  current
203    Gargi  2000    21/01/2015  current
```

Print records whose acc. type is 'saving'

```
root@MUM084:~/Desktop# awk '$5 ~/saving/' bank.lst
102    Anil   10000020/05/2011  saving
104    Ram    10000015/08/2010  saving
105    Jyotsna        5000  16/06/2012  saving
107    Vishal         14500030/11/2011  saving
201    Bina   3000    11/03/2010  saving
204    Hina   30000   14/02/2014  saving
301    Nikhil         7777  8/9/1999      saving
```

Print records whose acc. type is not 'saving'

```
root@MUM084:~/Desktop# awk '$5 !~/saving/' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
103    Naman      0       20/08/2009  current
106    Mukesh         14000020/12/2009  Current
108    Chirag         0       15/12/2012  Current
109    Arya   16000014/12/2010  Current
110    Priya  130     16/11/2009  Saving
202    Diya   4000    13/04/2018  Saving
203    Gargi  2000    21/01/2015  current
205    Kalpana        4000  8/9/2007      Current
```

Print records not ending with character 'g'

```
root@MUM084:~/Desktop# awk '$5 !~/g$/' bank.lst

101    ADITYA     0       14/11/2000  CURRENT
103    Naman      0       20/08/2009  current
106    Mukesh     1400020/12/2009  Current
108    Chirag     0       15/12/2012  Current
109    Arya  1600014/12/2010  Current
203    Gargi 2000  21/01/2015  current
205    Kalpana    4000  8/9/2007    Current
```

Print records ending with character 'g'

```
root@MUM084:~/Desktop# awk '$5 ~/g$/' bank.lst

102    Anil   1000020/05/2011  saving
104    Ram    1000015/08/2010  saving
105    Jyotsna    5000  16/06/2012  saving
107    Vishal     1450030/11/2011  saving
110    Priya  130    16/11/2009  Saving
201    Bina   3000  11/03/2010  saving
202    Diya   4000  13/04/2018  Saving
204    Hina   3000014/02/2014  saving
301    Nikhil     7777  8/9/1999    saving
```

Print names of customers having saving account.

```
root@MUM084:~/Desktop# awk '$5 ~/saving/{print $2}' bank.lst
Anil
Ram
Jyotsna
Vishal
Bina
Hina
Nikhil
```

Display records whose account type is current or Current

```
root@MUM084:~/Desktop# awk '$5~/[Cc]urrent/' bank.lst
103    Naman      0       20/08/2009  current
106    Mukesh     1400020/12/2009  Current
108    Chirag     0       15/12/2012  Current
109    Arya  1600014/12/2010  Current
203    Gargi 2000  21/01/2015  current
205    Kalpana    4000  8/9/2007    Current
```

Display those records whose account type field end ith letter t or T

```
root@MUM084:~/Desktop# awk '$5~/[Tt]$/' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
103    Naman      0       20/08/2009  current
106    Mukesh     14000   20/12/2009  Current
108    Chirag     0       15/12/2012  Current
109    Arya  16000 14/12/2010  Current
203    Gargi 2000  21/01/2015  current
205    Kalpana    4000  8/9/2007      Current
```

Display account number, name and balance of those records whose account type field end ith letter t or T

```
root@MUM084:~/Desktop# awk '$5~/[Tt]$/{print $1, $2, $3}' bank.lst
101 ADITYA 0
103 Naman 0
106 Mukesh 14000
108 Chirag 0
109 Arya 16000
203 Gargi 2000
205 Kalpana 4000
```

Display records from record number 3 to record number 7.

```
root@MUM084:~/Desktop# awk 'NR>=3 && NR<=7 {print $1, $2, $3}' bank.lst
103 Naman 0
104 Ram 10000
105 Jyotsna 5000
106 Mukesh 14000
107 Vishal 14500
```

Display records from record no. 2 to record no.8.

```
root@MUM084:~/Desktop# awk 'NR==2, NR==8 {print $1, $2, $3}' bank.lst
102 Anil 10000
103 Naman 0
104 Ram 10000
105 Jyotsna 5000
106 Mukesh 14000
107 Vishal 14500
108 Chirag 0
```

Display acc. No, name and balance of records having record no. 2 or record no.8 along with record no.

root@MUM084:~/Desktop# awk 'NR==2 || NR==8 {print NR, $1, $2, $3}' bank.lst
2 102 Anil 10000
8 108 Chirag 0

display name with record no. Having record no less than 2 and more than 8.
root@MUM084:~/Desktop# awk 'NR<2 || NR>8 {print NR, $2}' bank.lst
1 ADITYA
9 Arya
10 Priya
11 Bina
12 Diya
13 Gargi
14 Hina
15 Kalpana
16 Nikhil

Create text.lst file as shown below:
root@MUM084:~/Desktop# cat > text.lst
This is unix operating system
we are studying AWK scripts
It appears to be very interesting

Print number of fields in each line /record of the file text.lst
root@MUM084:~/Desktop# awk '{print NF}' text.lst
5
5
6

Print last field of each line in text.lst
root@MUM084:~/Desktop# awk '{print $NF}' text.lst
system
scripts
interesting

Print last field of each line in bank.lst
root@MUM084:~/Desktop# awk '{print $NF}' bank.lst
CURRENT
saving
current
saving
saving
Current
saving
Current
Current
Saving

saving
Saving
current
saving
Current
saving

Print all records having A at the beginning of second field

```
root@MUM084:~/Desktop# awk '$2 ~ "^A"' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
102    Anil   1000020/05/2011  saving
109    Arya  1600014/12/2010  Current
```

Print all records having A at the beginning of second field and g at the end of fifth field

```
root@MUM084:~/Desktop# awk '$2 ~ "^A" && $5 ~ "g$"' bank.lst
102    Anil   1000020/05/2011  saving
```

Print records whose date of opening starts with 14 or the year ends with 12.

```
root@MUM084:~/Desktop# awk '$4 ~ "^14" || $4~"12$"' bank.lst
101    ADITYA     0       14/11/2000  CURRENT
105    Jyotsna        5000  16/06/2012  saving
108    Chirag         0       15/12/2012  Current
109    Arya  1600014/12/2010  Current
204    Hina  3000014/02/2014  saving
```

Print records whose date of opening is 20 and the year is 09.

```
root@MUM084:~/Desktop# awk '$4 ~ /^20.*09/' bank.lst
103    Naman          0       20/08/2009  current
106    Mukesh         1400020/12/2009  Current
```

Print records whose date of opening is 20 and the year is 09 and whose name starts with N

```
root@MUM084:~/Desktop# awk '$4 ~ /^20.*09/ && $2 ~/^N/' bank.lst
103    Naman          0       20/08/2009  current
```

Print all lines whose length is more than 32

```
root@MUM084:~/Desktop# awk 'length($0) > 32' bank.lst
105    Jyotsna        5000  16/06/2012  saving
106    Mukesh         1400020/12/2009  Current
107    Vishal         1450030/11/2011  saving
109    Arya  1600014/12/2010  Current
```

203 Gargi 2000 21/01/2015 current
205 Kalpana 4000 8/9/2007 Current

Performing arithmatic operation:
Print customer name, balance, date and 5% interest on balance

root@MUM084:~/Desktop# awk '$5 == "saving" { printf "%20s %d %20s %f \n",$2, $3, $4, $3*0.05}' bank.lst
Anil 10000 20/05/2011 500.000000
Ram 10000 15/08/2010 500.000000
Jyotsna 5000 16/06/2012 250.000000
Vishal 14500 30/11/2011 725.000000
Bina 3000 11/03/2010 150.000000
Hina 30000 14/02/2014 1500.000000
Nikhil 7777 8/9/1999 388.850000

Use of BEGIN and END keywords:

If u have something to print before processing the first line, for e.g. a heading then the BEGIN section is used.

Similarly, the END section is useful in printing something after processing is over.

They are optional.

Syntactical form:

awk ' BEGIN{actions}

/pattern/ {actions}

END{actions}' file_name

Note: Opening curly brackets should be on the same line of BEGIN and END

root@MUM084:~/Desktop# awk '**BEGIN{**
printf "Records in the bank are :\n"
}
{print $1, $2, $3}' bank.lst

Records in the bank are :
101 ADITYA 0
102 Anil 10000
103 Naman 0
104 Ram 10000

105 Jyotsna 5000
106 Mukesh 14000
107 Vishal 14500
108 Chirag 0
109 Arya 16000
110 Priya 130
201 Bina 3000
202 Diya 4000
203 Gargi 2000
204 Hina 30000
205 Kalpana 4000
301 Nikhil 7777

root@MUM084:~/Desktop# awk '**BEGIN{**
printf "Records in the bank are :\n"
}
{print $1, $2, $3}
> **END{print "\n we displayed all records"}**' bank.lst
Records in the bank are :
101 ADITYA 0
102 Anil 10000
103 Naman 0
104 Ram 10000
105 Jyotsna 5000
106 Mukesh 14000
107 Vishal 14500
108 Chirag 0
109 Arya 16000
110 Priya 130
201 Bina 3000
202 Diya 4000
203 Gargi 2000
204 Hina 30000
205 Kalpana 4000
301 Nikhil 7777

we displayed all records

Storing commands in file and using them:

Store the command to calculate total balance of all accounts and printing it at the end. Use this command on bank.lst file

root@MUM084:~/Desktop# cat >totalbal.awk

```
{total+=$3}
END{print"total balance is = ", total}
root@MUM084:~/Desktop# awk -f totalbal.awk bank.lst
total balance is =  120407
```

What will happen if END section is not used in the above command?

i.e. totalbal.awk file is modified as:

```
{total+=$3}
{print"total balance is = ", total}
```

and then it is executed on bank.lst. We get following output:

```
root@MUM084:~/Desktop# awk -f totalbal.awk bank.lst
total balance is =  0
total balance is =  10000
total balance is =  10000
total balance is =  20000
total balance is =  25000
total balance is =  39000
total balance is =  53500
total balance is =  53500
total balance is =  69500
total balance is =  69630
total balance is =  72630
total balance is =  76630
total balance is =  78630
total balance is =  108630
total balance is =  112630
total balance is =  120407
```

Create file countrec.awk which contains actions to count no. Of records, to calculate total balance and average balance. Apply commands in this file to bank.lst

```
root@MUM084:~/Desktop# cat > countrec.awk
BEGIN{printf "Records are: \n"
}
{
print $0
c++
sum+=$3
}
END{printf "\n Number of records are: %d", c
printf "\n Total balance is  %d", sum
printf "\n Average balance is " %f", sum/c
```

```
}
root@MUM084:~/Desktop# awk -f countrec.awk bank.lst
Records are:
101    ADITYA     0       14/11/2000  CURRENT
102    Anil   1000020/05/2011  saving
103    Naman      0       20/08/2009  current
104    Ram    1000015/08/2010  saving
105    Jyotsna        5000  16/06/2012  saving
106    Mukesh     1400020/12/2009  Current
107    Vishal     1450030/11/2011  saving
108    Chirag     0       15/12/2012  Current
109    Arya   1600014/12/2010  Current
110    Priya  130    16/11/2009  Saving
201    Bina   3000   11/03/2010  saving
202    Diya   4000   13/04/2018  Saving
203    Gargi  2000   21/01/2015  current
204    Hina   3000014/02/2014  saving
205    Kalpana        4000  8/9/2007      Current
301    Nikhil         7777  8/9/1999      saving

 Number of records are: 16
 Total balance is  120407
 Average balance is  7525.437500root@MUM084:~/Desktop#
```

Create file addnonzero.awk which contains actions to add only nonzero balance, display total balance and average balance. Apply commands in this file to bank.lst

```
root@MUM084:~/Desktop# cat > addnonzero.awk
$3==0{next}
{total+=$3
count++
}
END{avg=total/count
printf"\n Total is : %d", total
printf"\n average is: %d", avg
printf"\n no. of customers is %d", count}
root@MUM084:~/Desktop# awk -f addnonzero.awk bank.lst

 Total is : 120407
 average is: 9262
 no. of customers is 13
```

Update nonzero.awk file such that it adds balance of those customers whose account type is not 'saving', displays their average and no of such records

Hence the nonzero.awk now becomes:

```
$5 ~/^s/{next}
{total+=$3
count++
}
END{avg=total/count
printf"\n Total is : %d", total
printf"\n average is: %d", avg
printf"\n no. of customers is %d", count}
```

```
root@MUM084:~/Desktop# awk -f addnonzero.awk bank.lst

 Total is : 40130
 average is: 4458
 no. of customers is 9
```

Using if.. else

command to calculate 5% interest if balance more than 10000 else interest is 6%

```
root@MUM084:~/Desktop# cat >interest
{if ($3 > 10000) print "interest = " $3*0.05;
else print "interest = " $3*0.06}
```

```
root@MUM084:~/Desktop# awk -f interest bank.lst
interest = 0
interest = 600
interest = 0
interest = 600
interest = 300
interest = 700
interest = 725
interest = 0
interest = 800
interest = 7.8
interest = 180
interest = 240
interest = 120
interest = 1500
interest = 240
```

interest = 466.62

Write command to calculate 5% interest if account type is 'current' else interest is 6%

interest file is updated as below:

{if ($5~/current/) print "interest = " $3*0.05;
else print "interest = " $3*0.06}

root@MUM084:~/Desktop# awk -f interest bank.lst
interest = 0
interest = 600
interest = 0
interest = 600
interest = 300
interest = 840
interest = 870
interest = 0
interest = 960
interest = 7.8
interest = 180
interest = 240
interest = 100
interest = 1800
interest = 240
interest = 466.62

Write command to calculate 5% interest if account type is 'current' or 'Current' else interest is 6%

interest file is updated as below:

{if ($5~/[Cc]urrent/) print "interest = " $3*0.05;
else print "interest = " $3*0.06}

root@MUM084:~/Desktop# awk -f interest bank.lst
interest = 0
interest = 600
interest = 0
interest = 600
interest = 300
interest = 700
interest = 870
interest = 0
interest = 800

interest = 7.8
interest = 180
interest = 240
interest = 100
interest = 1800
interest = 200
interest = 466.62
root@MUM084:~/Desktop#

Write command to calculate 5% interest if account type is not 'current' or 'Current' else interest is 6%

interest file is updated as below:

{if ($5!~/[Cc]urrent/) print "interest = " $3*0.05;
else print "interest = " $3*0.06}
root@MUM084:~/Desktop# awk -f interest bank.lst
interest = 0
interest = 500
interest = 0
interest = 500
interest = 250
interest = 840
interest = 725
interest = 0
interest = 960
interest = 6.5
interest = 150
interest = 200
interest = 120
interest = 1500
interest = 240
interest = 388.85
root@MUM084:~/Desktop#