BASHAVATHINI

Harsha Vardhan

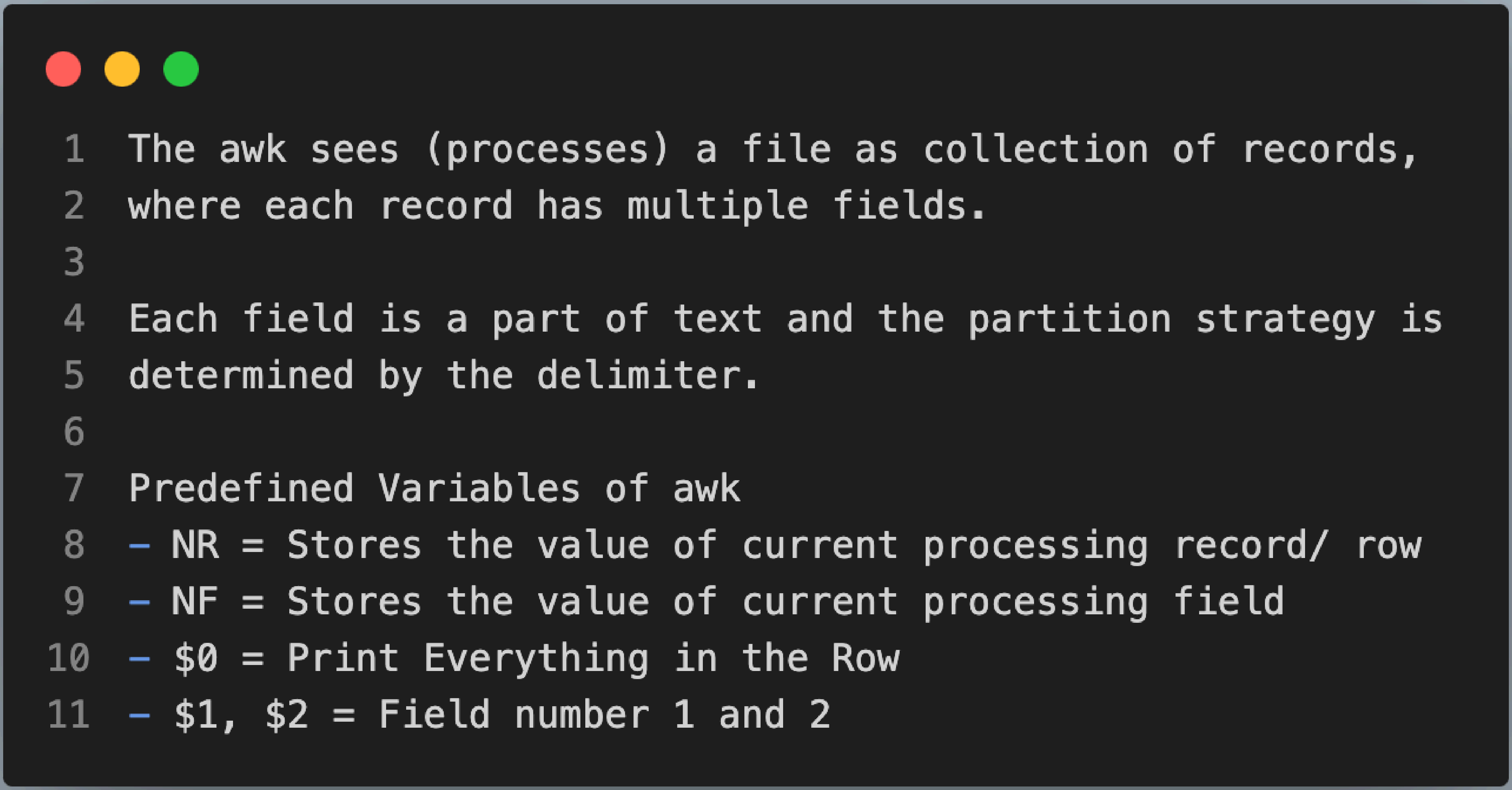
DEVOPS ENGINEER

Master awk command in 15 slides

by Harsha Vardhan Bashavathini

→

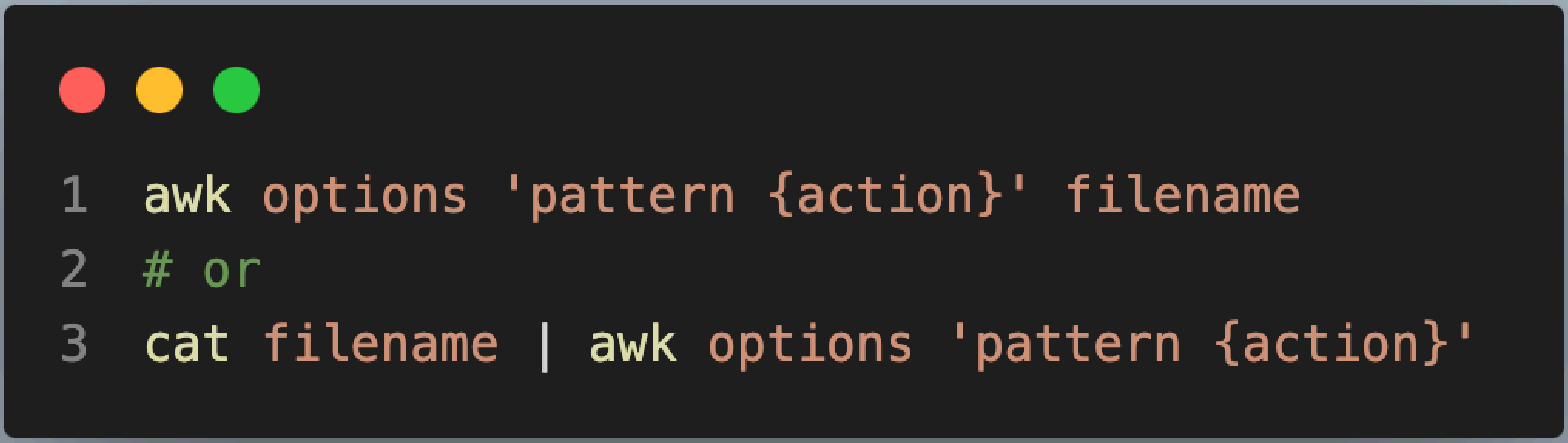
Introduction



```
The awk sees (processes) a file as collection of records,
where each record has multiple fields.

Each field is a part of text and the partition strategy is
determined by the delimiter.

Predefined Variables of awk
- NR = Stores the value of current processing record/ row
- NF = Stores the value of current processing field
- $0 = Print Everything in the Row
- $1, $2 = Field number 1 and 2
```

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

Syntax of awk command

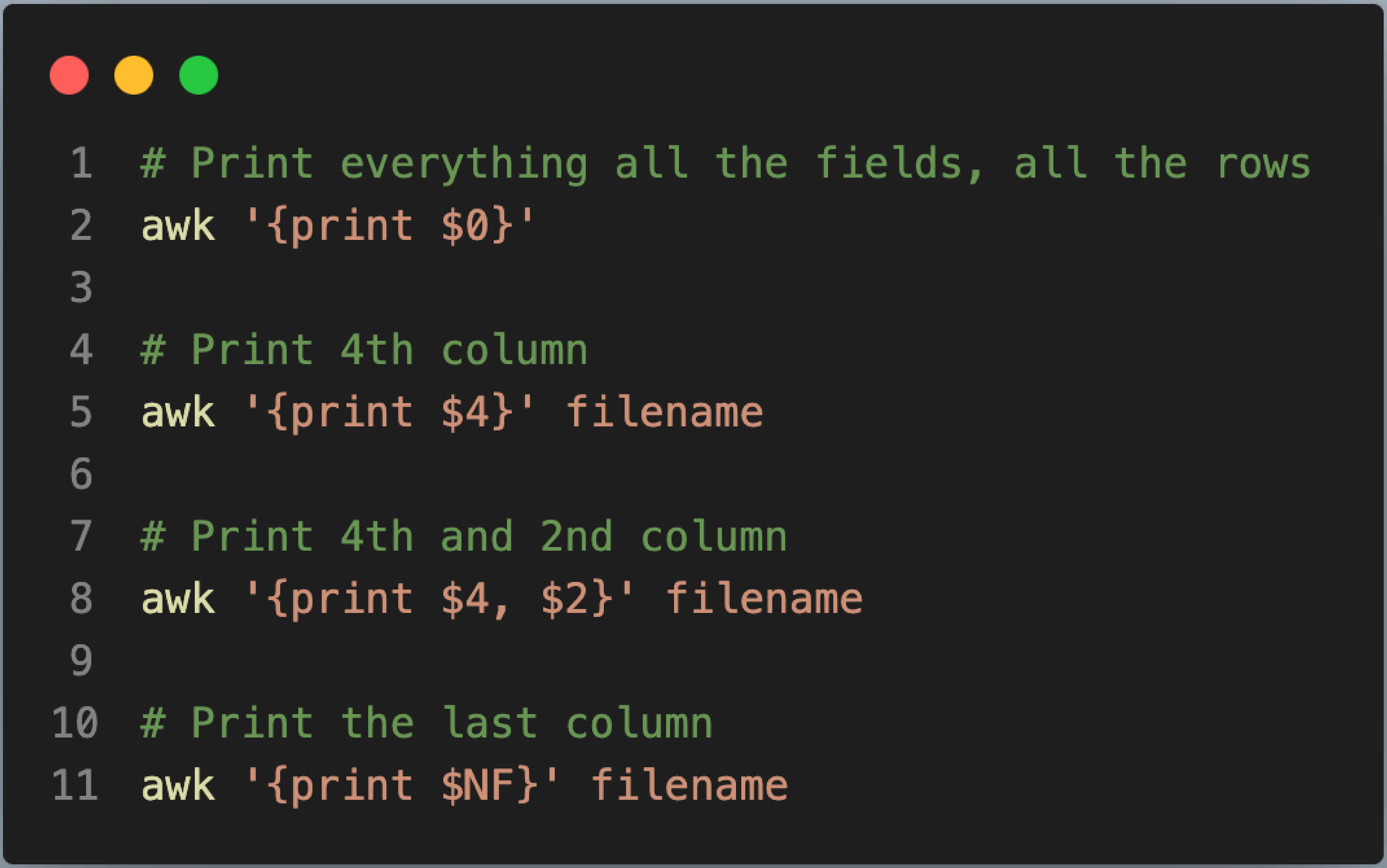


```
awk options 'pattern {action}' filename
# or
cat filename | awk options 'pattern {action}'
```

Harsha Vardhan Bashavathini

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

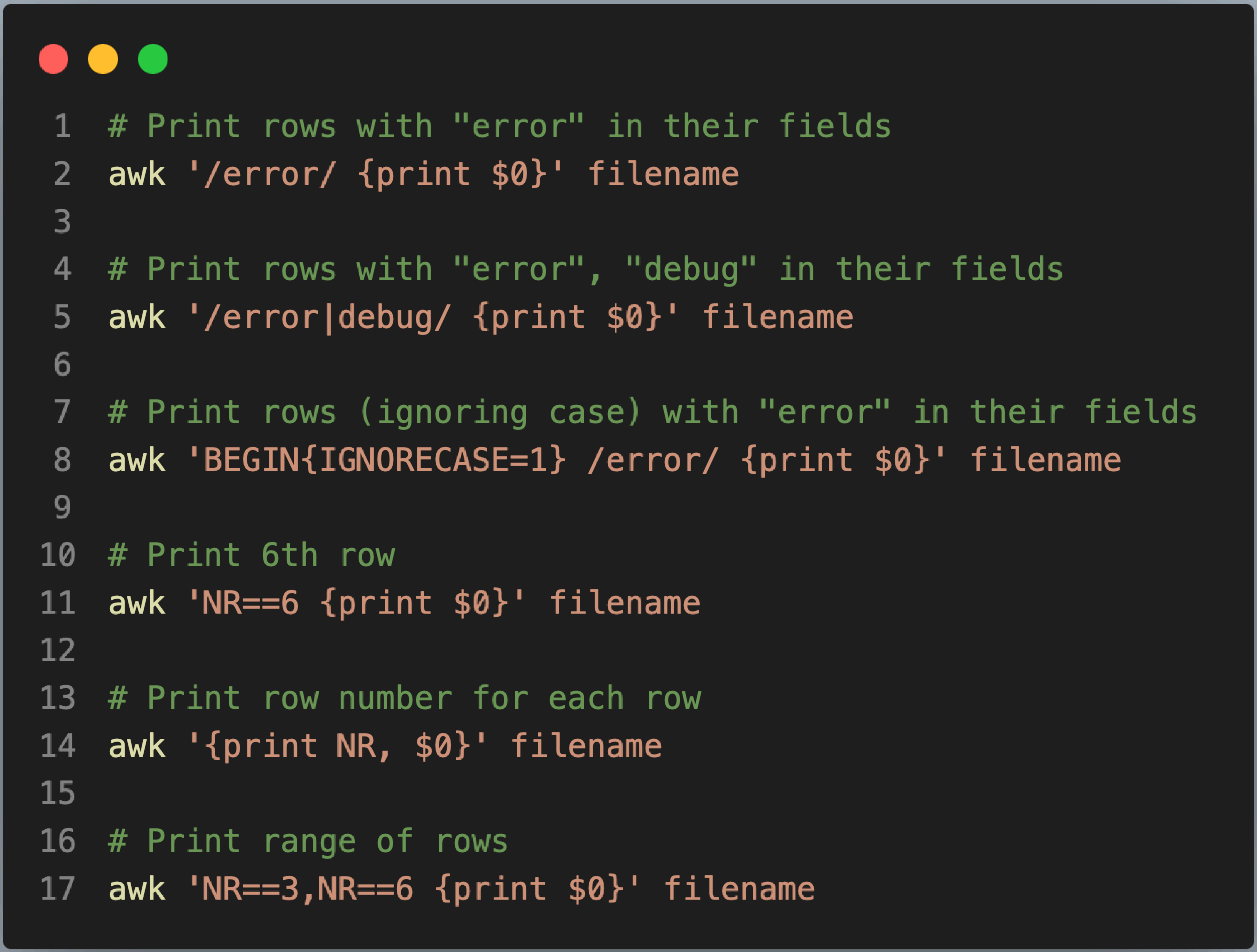
Print a specific column



```
# Print everything all the fields, all the rows
awk '{print $0}'

# Print 4th column
awk '{print $4}' filename

# Print 4th and 2nd column
awk '{print $4, $2}' filename

# Print the last column
awk '{print $NF}' filename
```

Harsha Vardhan Bashavathini

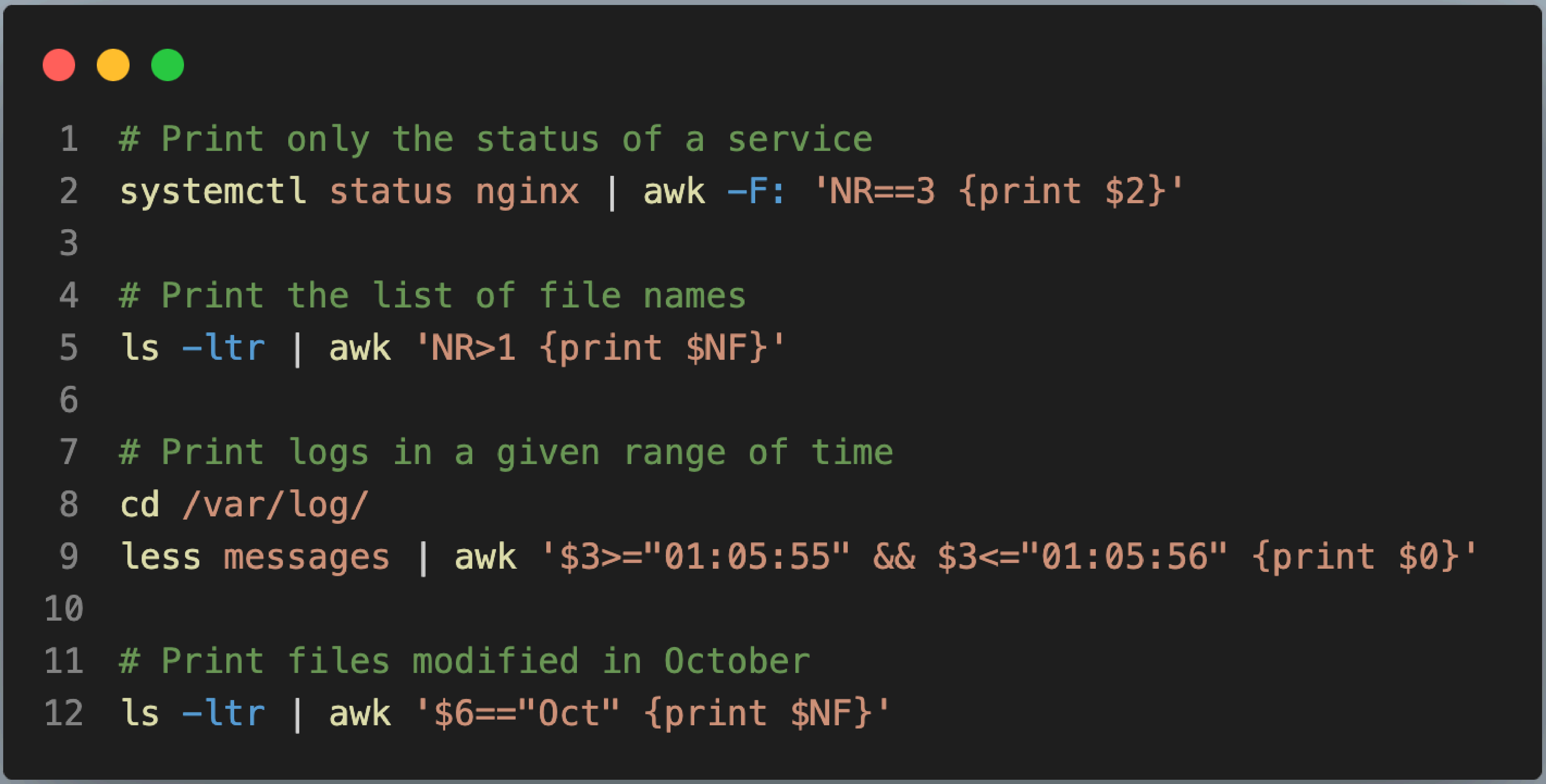
Print a specific row



```
# Print rows with "error" in their fields
awk '/error/ {print $0}' filename

# Print rows with "error", "debug" in their fields
awk '/error|debug/ {print $0}' filename

# Print rows (ignoring case) with "error" in their fields
awk 'BEGIN{IGNORECASE=1} /error/ {print $0}' filename

# Print 6th row
awk 'NR==6 {print $0}' filename

# Print row number for each row
awk '{print NR, $0}' filename

# Print range of rows
awk 'NR==3,NR==6 {print $0}' filename
```

Harsha Vardhan Bashavathini

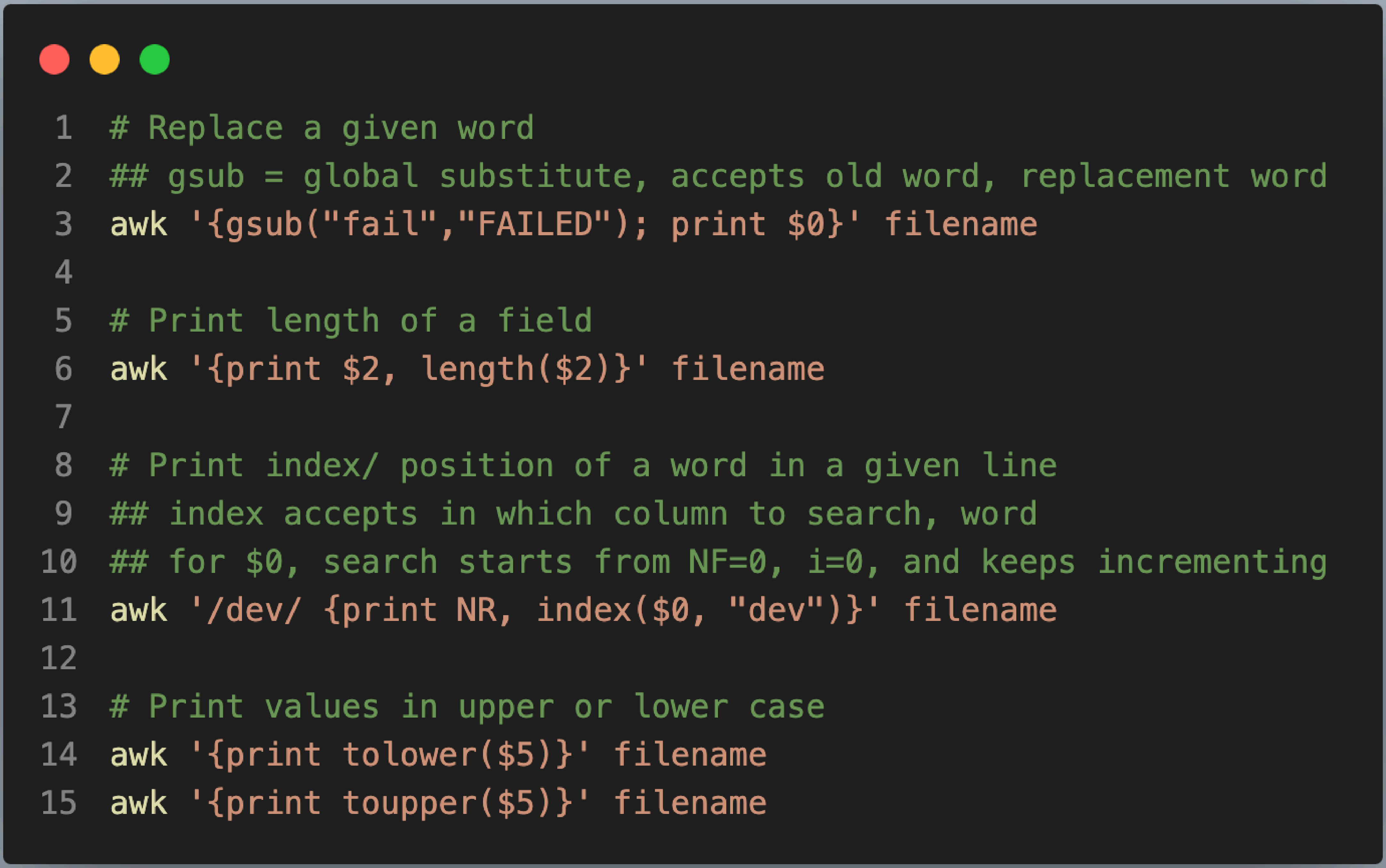
awk with other Commands



```
# Print only the status of a service
systemctl status nginx | awk -F: 'NR==3 {print $2}'

# Print the list of file names
ls -ltr | awk 'NR>1 {print $NF}'

# Print logs in a given range of time
cd /var/log/
less messages | awk '$3>="01:05:55" && $3<="01:05:56" {print $0}'

# Print files modified in October
ls -ltr | awk '$6=="Oct" {print $NF}'
```

Harsha Vardhan Bashavathini

In-built commands

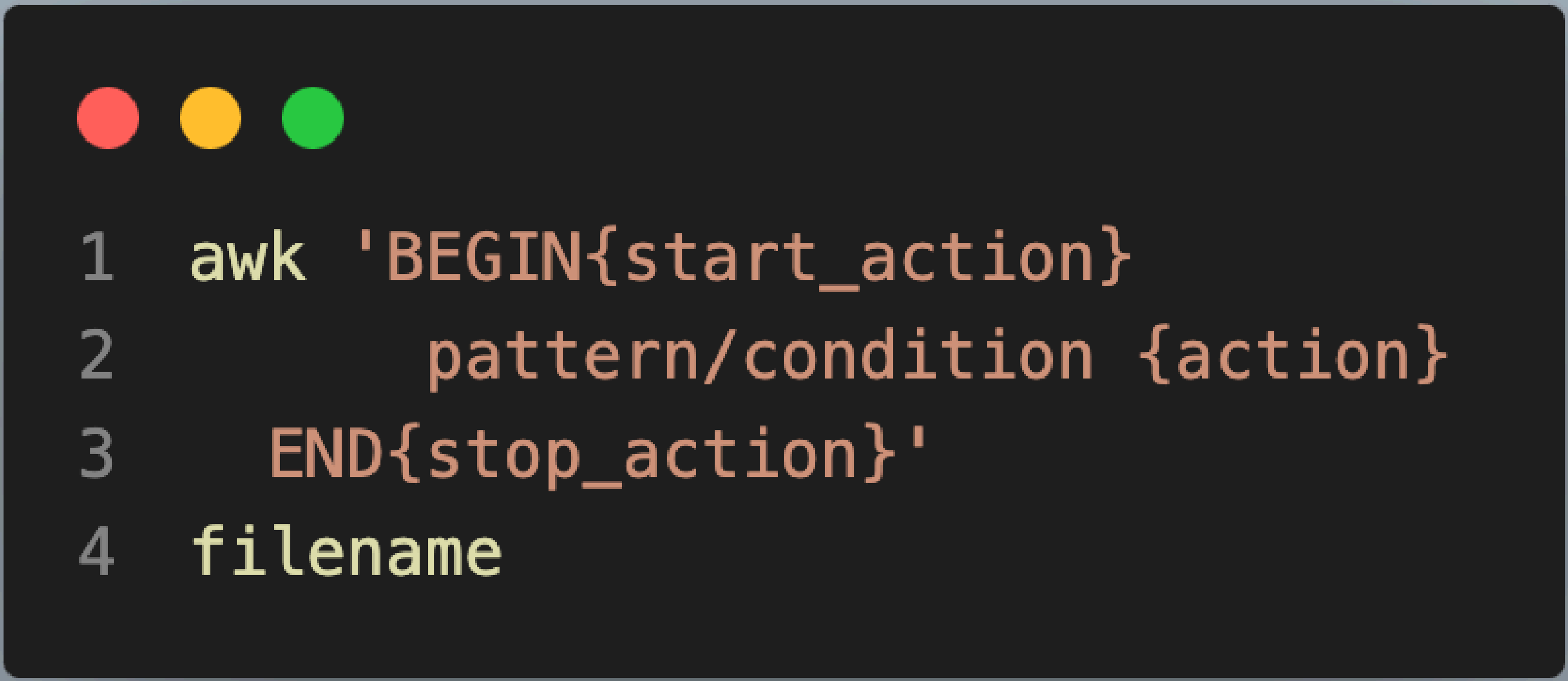


```
# Replace a given word
## gsub = global substitute, accepts old word, replacement word
awk '{gsub("fail","FAILED"); print $0}' filename

# Print length of a field
awk '{print $2, length($2)}' filename

# Print index/ position of a word in a given line
## index accepts in which column to search, word
## for $0, search starts from NF=0, i=0, and keeps incrementing
awk '/dev/ {print NR, index($0, "dev")}' filename

# Print values in upper or lower case
awk '{print tolower($5)}' filename
awk '{print toupper($5)}' filename
```

Harsha Vardhan Bashavathini

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

Syntax of using BEGIN and END

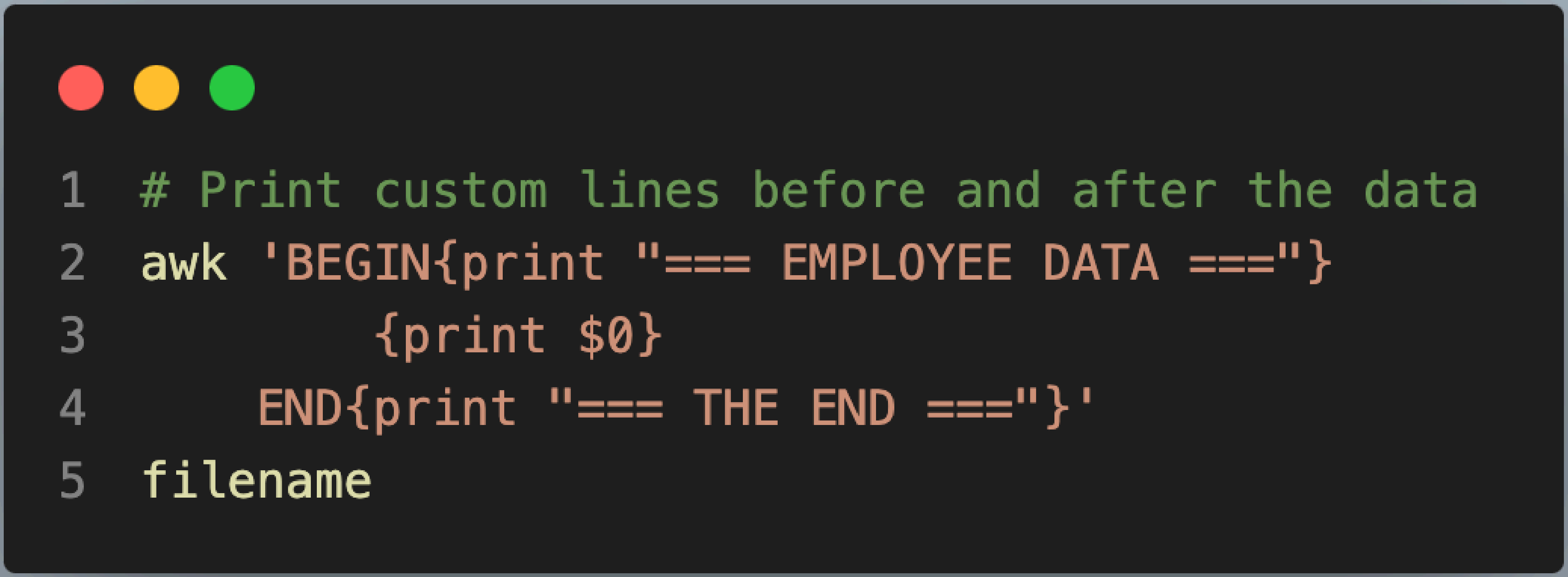


```
awk 'BEGIN{start_action}
      pattern/condition {action}
  END{stop_action}'
filename
```

Harsha Vardhan Bashavathini

Usecase #1

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

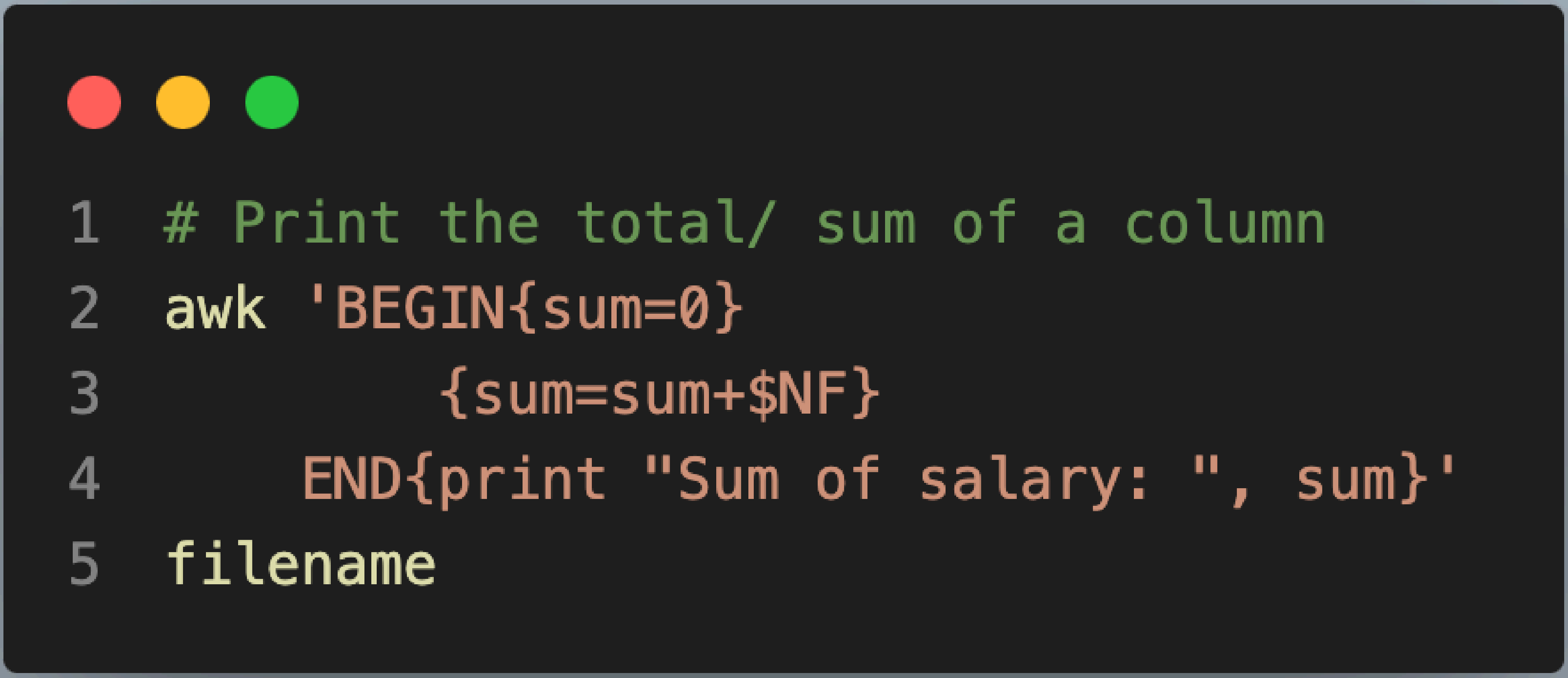


```
# Print custom lines before and after the data
awk 'BEGIN{print "=== EMPLOYEE DATA ==="}
           {print $0}
      END{print "=== THE END ==="}'
filename
```

Harsha Vardhan Bashavathini

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

Usecase #2

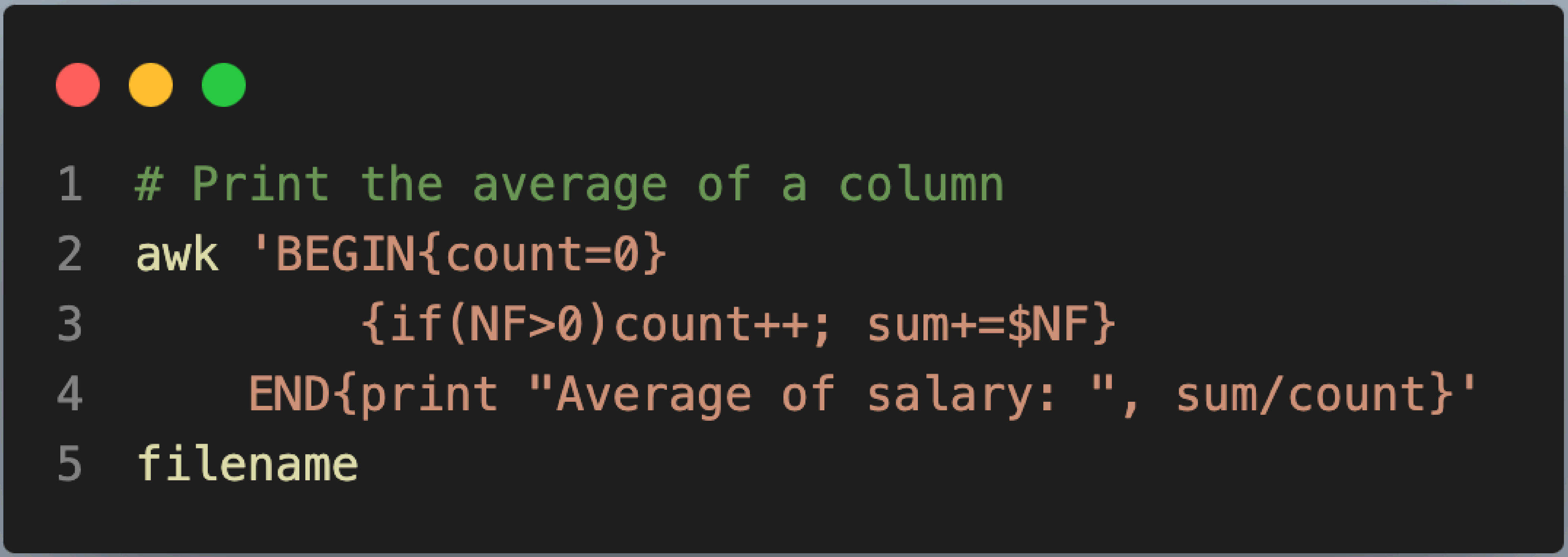


```
# Print the total/ sum of a column
awk 'BEGIN{sum=0}
        {sum=sum+$NF}
    END{print "Sum of salary: ", sum}'
filename
```

Harsha Vardhan Bashavathini

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

Usecase #3

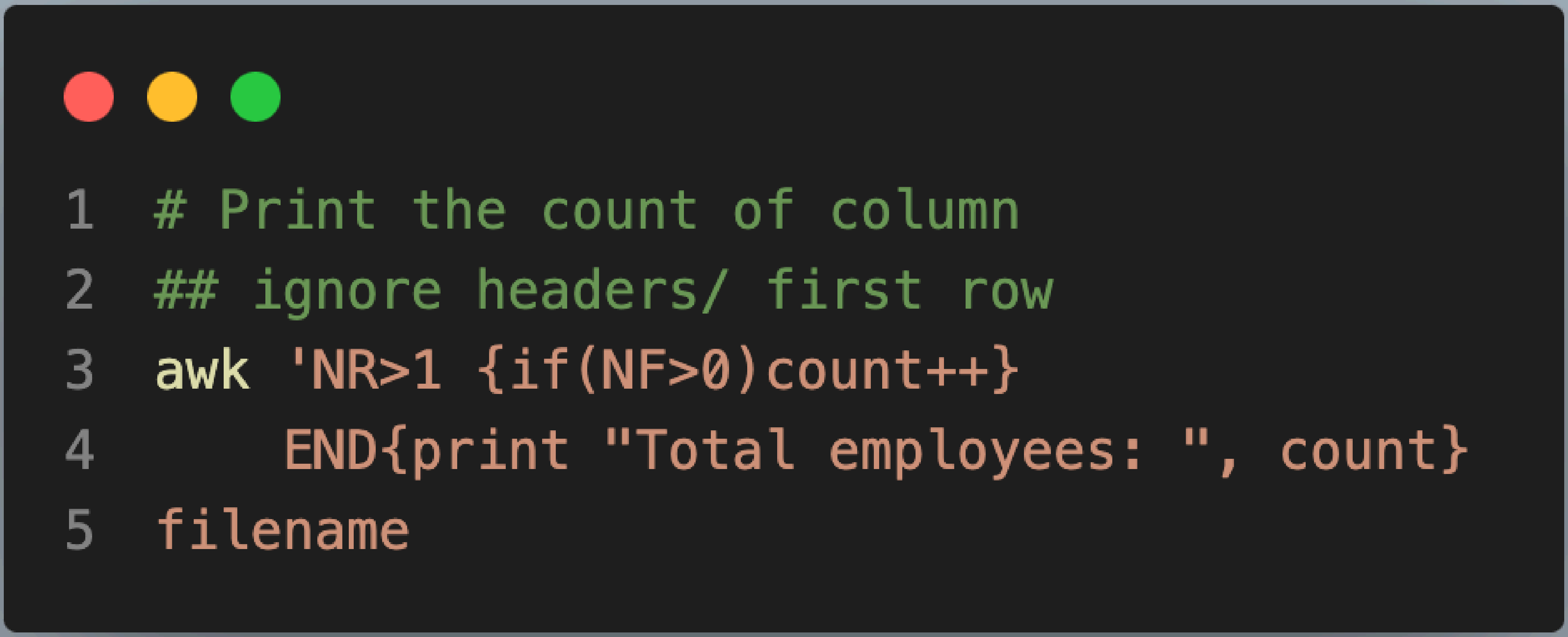


```
# Print the average of a column
awk 'BEGIN{count=0}
        {if(NF>0)count++; sum+=$NF}
    END{print "Average of salary: ", sum/count}'
filename
```

Harsha Vardhan Bashavathini

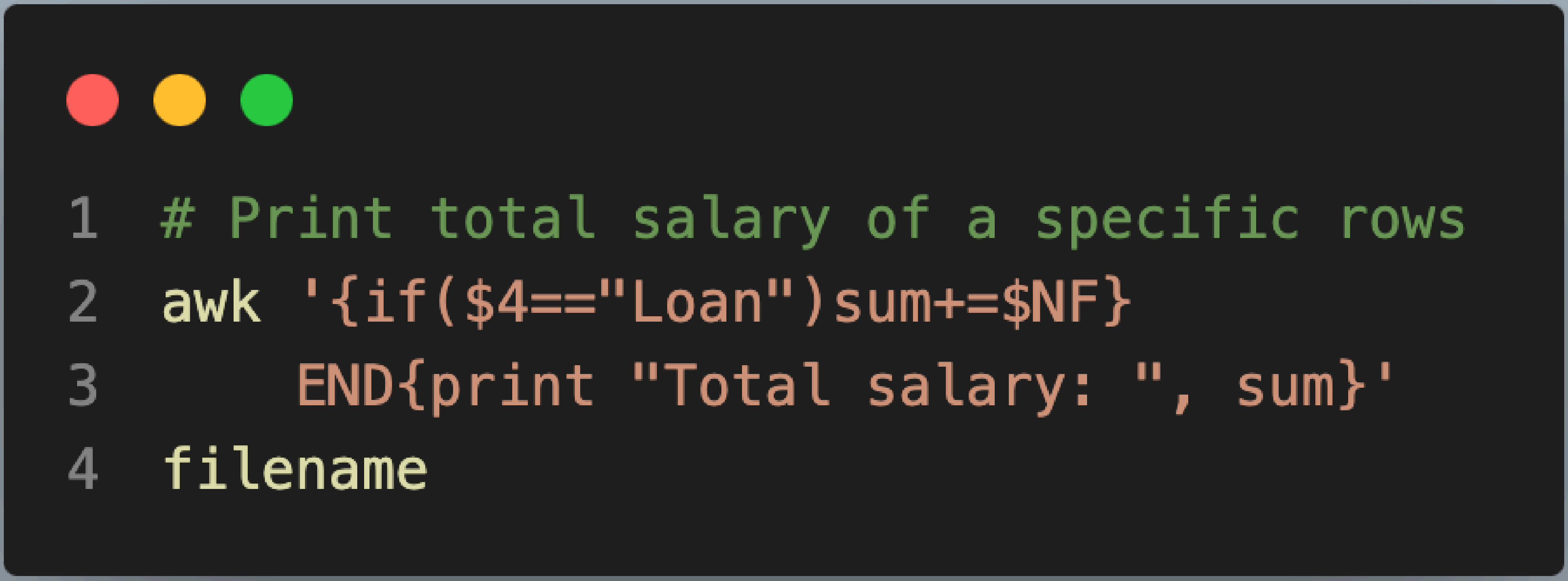
Usecase #4

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER



```
# Print the count of column
## ignore headers/ first row
awk 'NR>1 {if(NF>0)count++}
    END{print "Total employees: ", count}
filename
```

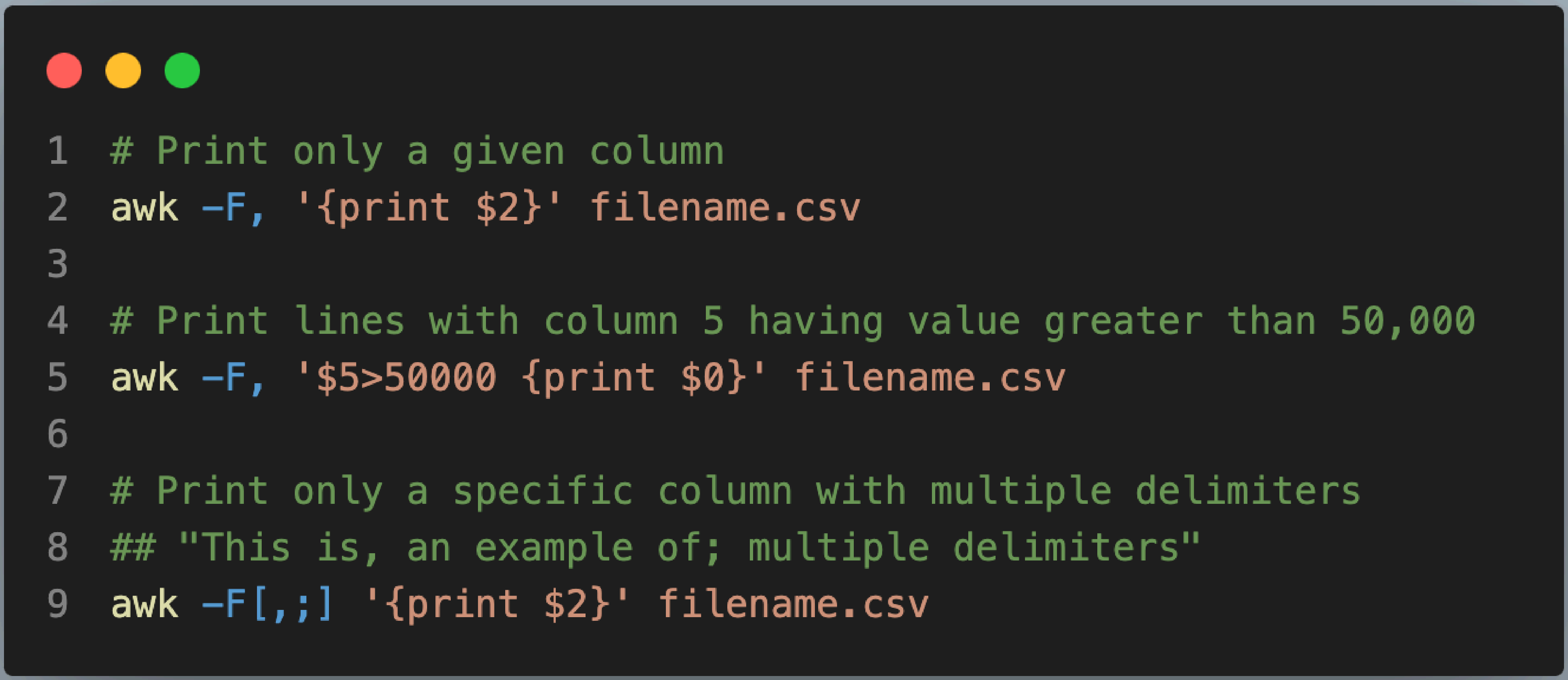
Usecase #5



```
# Print total salary of a specific rows
awk '{if($4=="Loan")sum+=$NF}
    END{print "Total salary: ", sum}'
filename
```

Harsha Vardhan Bashavathini

Usecase #6



```
# Print only a given column
awk -F, '{print $2}' filename.csv

# Print lines with column 5 having value greater than 50,000
awk -F, '$5>50000 {print $0}' filename.csv

# Print only a specific column with multiple delimiters
## "This is, an example of; multiple delimiters"
awk -F[,;] '{print $2}' filename.csv
```

Harsha Vardhan Bashavathini

BASHAVATHINI
Harsha Vardhan
DEVOPS ENGINEER

THE END

By:

Harsha Vardhan Bashavathini

LIKE & SHARE