Unix Lab 5
roll no 423111

1. Given a log file with the format:
   [Timestamp] [Log Level] [Module] [Message].

Write an awk command to extract only ERROR messages along with their timestamps.

2. Given a CSV file with tab-separated values.

ID	Math	Science	English
1	78	85	90
2	82	80	88
3	75	92	95

Write an awk script to compute the average of each subject.

3. Given a server log with IP addresses:

192.168.1.1 - - [17/Feb/2025:12:00:01] "GET /index.html"
192.168.1.2 - - [17/Feb/2025:12:05:23] "POST /login"
192.168.1.1 - - [17/Feb/2025:12:10:45] "GET /dashboard"
Write an awk script to count occurrences of each IP.

4. Given lines of text:

apple banana cherry
dog cat elephant
Write a sed command to swap the first and last words.

5. Given a file with duplicate words:

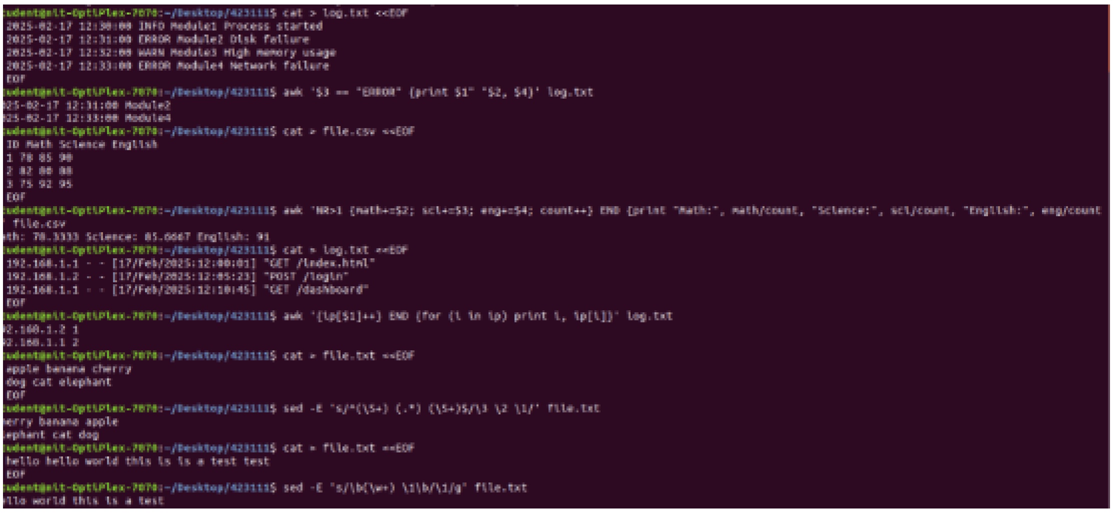
hello hello worldthis is is a test test
Write a sed command to remove consecutive duplicate words.

6. Given a file containing email addresses:

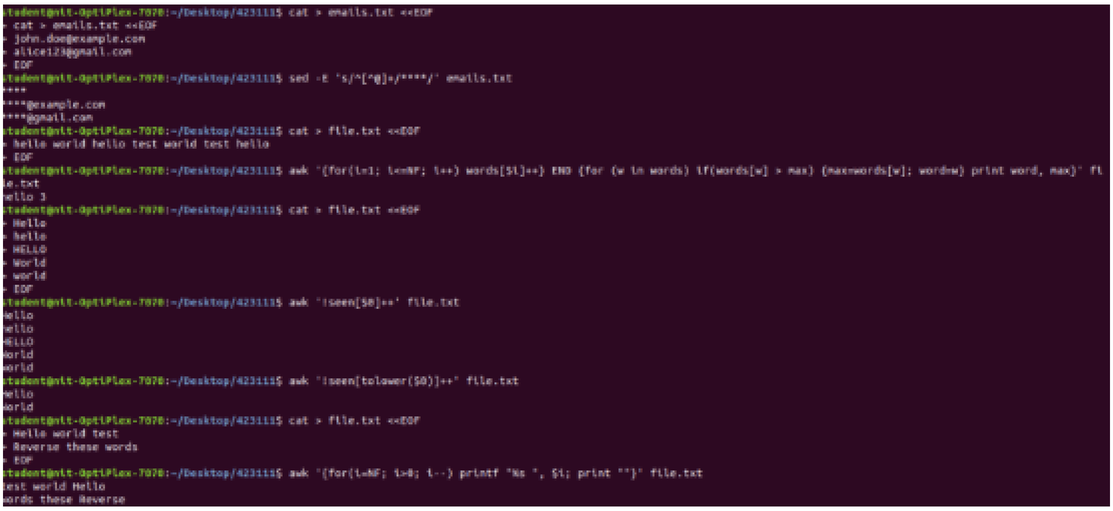
john.doe@example.com
alice123@gmail.com
Write a sed command to mask the usernames (before @).

7. Given a text file, find and print the most frequent word and its count.
8. Given a file with repeated lines in different cases, extract unique lines (case-insensitive and case-sensitive)
9. Given a file, reverse the order of words in each line.
10. Given a file, print the longest line and its length.
11. Given login records, extract usernames and sort by frequency.
12. Convert YYYY-MM-DD format to DD-MM-YYYY.
13. Normalize spacing by replacing multiple spaces with a single space.
14. Given a file with phone numbers, mask all but the last 4 digits.
15. Extract text inside parentheses.
16. Reverse the characters in each line of a file.
17. Extract the 3rd word from each line, if it exists.
18. Find and print lines containing words that are anagrams.
19. Remove lines that contain only numbers.
20. Identify the word that appears most frequently and its count.
21. Transform hello_world_example to helloWorldExample.
22. Add Line Numbers Only to Non-Empty Lines
23. Extract all email addresses from a file.
24. Replace All Digits with Corresponding Words: Convert numbers to words (1 → one, 2 → two).
25. Strip all HTML tags, leaving only plain text.
26. Replace Repeated Characters with a Single Instance
27. Extract Sentences Containing a Specific Word
28. Identify palindromes (words that read the same forward and backward) and wrap them in brackets
29. Detect and replace consecutive repeated words with a single instance.
30. Extract all unique words from a file, sort them alphabetically (ignoring case), and print them

Unix Lab 5
roll no 423111

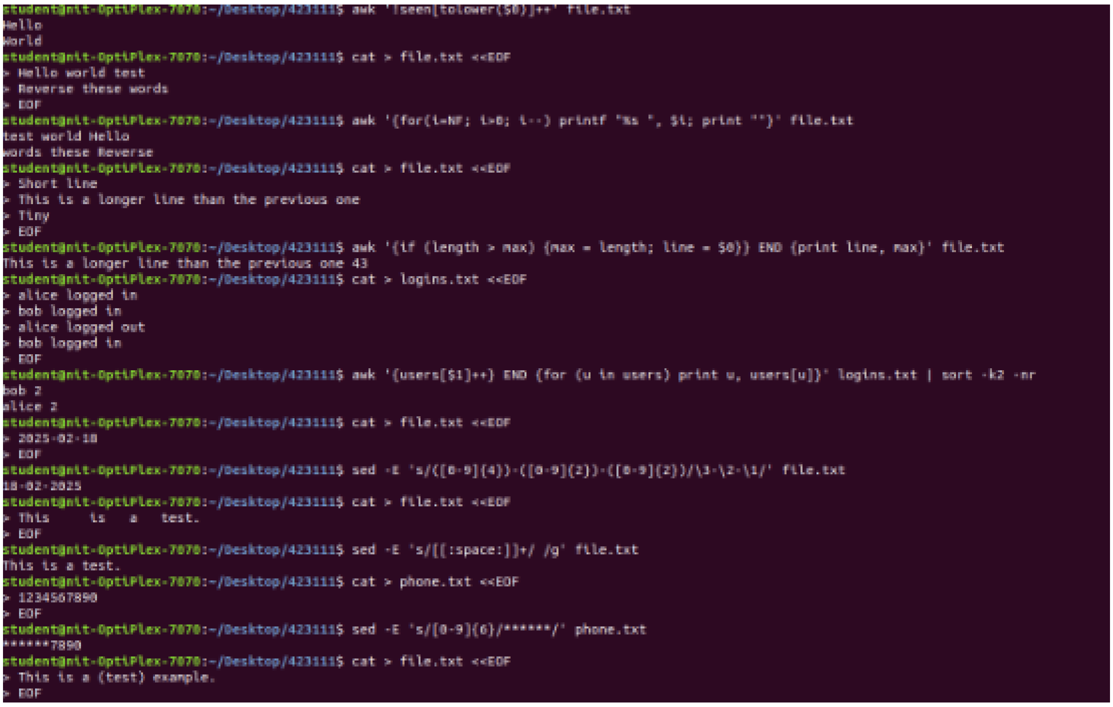


```
udent@nit-OptiPlex-7070:~/Desktop/423111$ cat > log.txt <<EOF
 2025-02-17 12:30:00 INFO Module1 Process started
 2025-02-17 12:31:00 ERROR Module2 Disk failure
 2025-02-17 12:32:00 WARN Module3 High memory usage
 2025-02-17 12:33:00 ERROR Module4 Network failure
 EOF
udent@nit-OptiPlex-7070:~/Desktop/423111$ awk '$3 == "ERROR" {print $1" "$2, $4}' log.txt
25-02-17 12:31:00 Module2
25-02-17 12:33:00 Module4
udent@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.csv <<EOF
 ID Math Science English
 1 78 85 90
 2 82 80 88
 3 75 92 95
 EOF
udent@nit-OptiPlex-7070:~/Desktop/423111$ awk 'NR>1 {math+=$2; sci+=$3; eng+=$4; count++} END {print "Math:", math/count, "Science:", sci/count, "English:", eng/count
' file.csv
th: 78.3333 Science: 85.6667 English: 91
udent@nit-OptiPlex-7070:~/Desktop/423111$ cat > log.txt <<EOF
 192.168.1.1 - - [17/Feb/2025:12:00:01] "GET /index.html"
 192.168.1.2 - - [17/Feb/2025:12:05:23] "POST /login"
 192.168.1.1 - - [17/Feb/2025:12:10:45] "GET /dashboard"
 EOF
udent@nit-OptiPlex-7070:~/Desktop/423111$ awk '{ip[$1]++} END {for (i in ip) print i, ip[i]}' log.txt
2.168.1.2 1
2.168.1.1 2
udent@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
 apple banana cherry
 dog cat elephant
 EOF
udent@nit-OptiPlex-7070:~/Desktop/423111$ sed -E 's/^(\S+) (.*) (\S+)$/\3 \2 \1/' file.txt
erry banana apple
ephant cat dog
udent@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
 hello hello world this is is a test test
 EOF
udent@nit-OptiPlex-7070:~/Desktop/423111$ sed -E 's/\b(\w+) \1\b/\1/g' file.txt
llo world this is a test
```

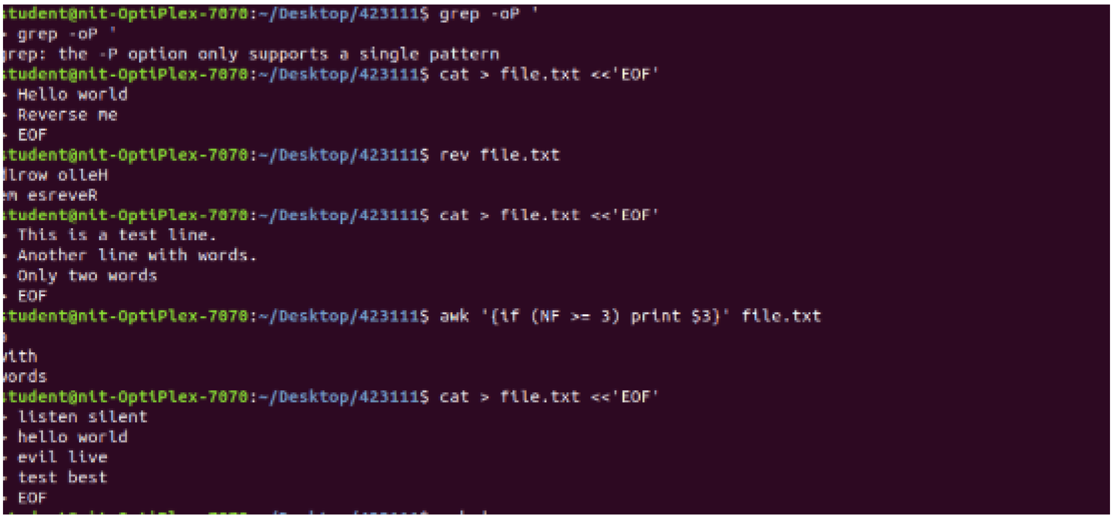


```
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > emails.txt <<EOF
> cat > emails.txt <<EOF
> john.doe@example.com
> alice123@gmail.com
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ sed -E 's/^[^@]+/****/' emails.txt
****
****@example.com
****@gmail.com
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> hello world hello test world test hello
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{for(i=1; i<=NF; i++) words[$i]++} END {for (w in words) if(words[w] > max) {max=words[w]; word=w} print word, max}' fi
le.txt
hello 3
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> Hello
> hello
> HELLO
> World
> world
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '!seen[$0]++' file.txt
Hello
hello
HELLO
World
world
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '!seen[tolower($0)]++' file.txt
Hello
World
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> Hello world test
> Reverse these words
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{for(i=NF; i>0; i--) printf "%s ", $i; print ""}' file.txt
test world Hello
words these Reverse
```

Unix Lab 5
roll no 423111

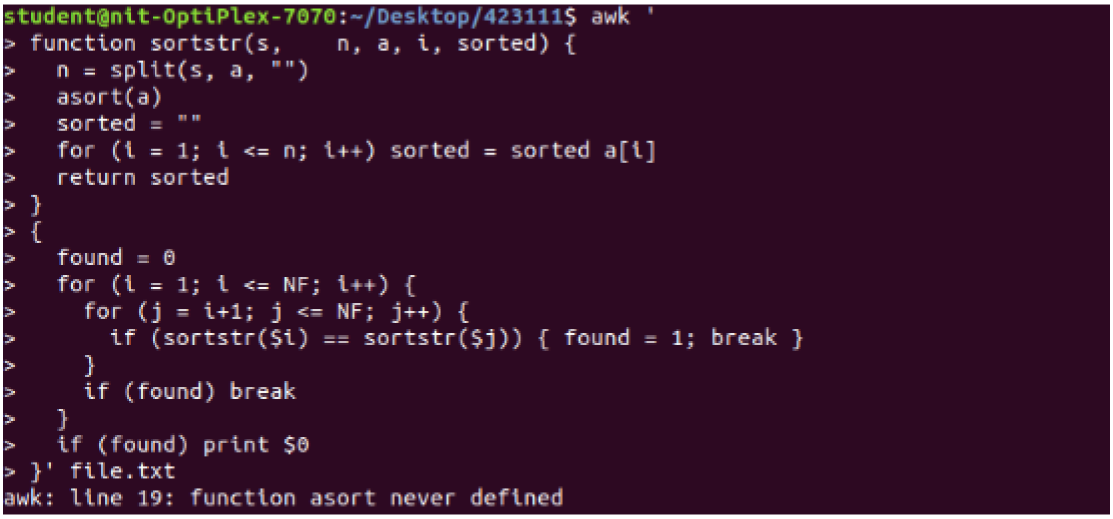


```
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '!seen[tolower($0)]++' file.txt
Hello
World
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> Hello world test
> Reverse these words
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{for(i=NF; i>0; i--) printf "%s ", $i; print ""}' file.txt
test world Hello
words these Reverse
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> Short line
> This is a longer line than the previous one
> Tiny
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{if (length > max) {max = length; line = $0}} END {print line, max}' file.txt
This is a longer line than the previous one 43
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > logins.txt <<EOF
> alice logged in
> bob logged in
> alice logged out
> bob logged in
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{users[$1]++} END {for (u in users) print u, users[u]}' logins.txt | sort -k2 -nr
bob 2
alice 2
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> 2025-02-18
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ sed -E 's/([0-9]{4})-([0-9]{2})-([0-9]{2})/\3-\2-\1/' file.txt
18-02-2025
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> This     is   a   test.
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ sed -E 's/[[:space:]]+/ /g' file.txt
This is a test.
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > phone.txt <<EOF
> 1234567890
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ sed -E 's/[0-9]{6}/******/' phone.txt
******7890
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<EOF
> This is a (test) example.
> EOF
```



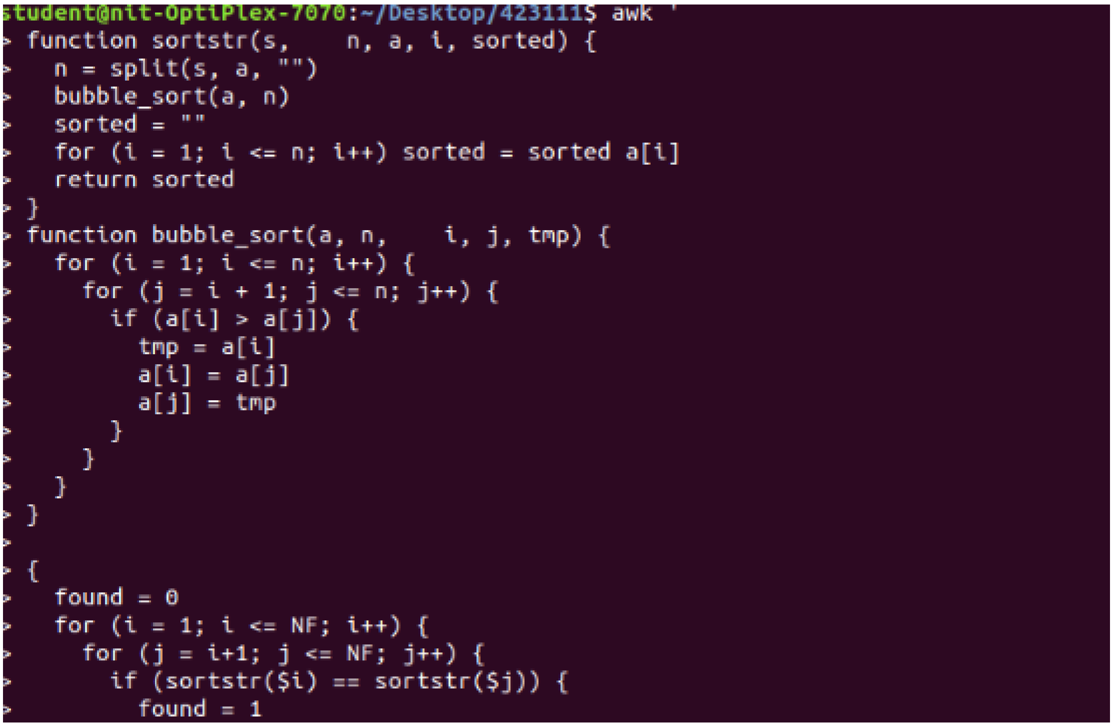
```
tudent@nit-OptiPlex-7070:~/Desktop/423111$ grep -oP '
 grep -oP '
rep: the -P option only supports a single pattern
tudent@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<'EOF'
 Hello world
 Reverse me
 EOF
tudent@nit-OptiPlex-7070:~/Desktop/423111$ rev file.txt
lrow olleH
n esreveR
tudent@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<'EOF'
 This is a test line.
 Another line with words.
 Only two words
 EOF
tudent@nit-OptiPlex-7070:~/Desktop/423111$ awk '{if (NF >= 3) print $3}' file.txt

ith
ords
tudent@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<'EOF'
 listen silent
 hello world
 evil live
 test best
 EOF
```



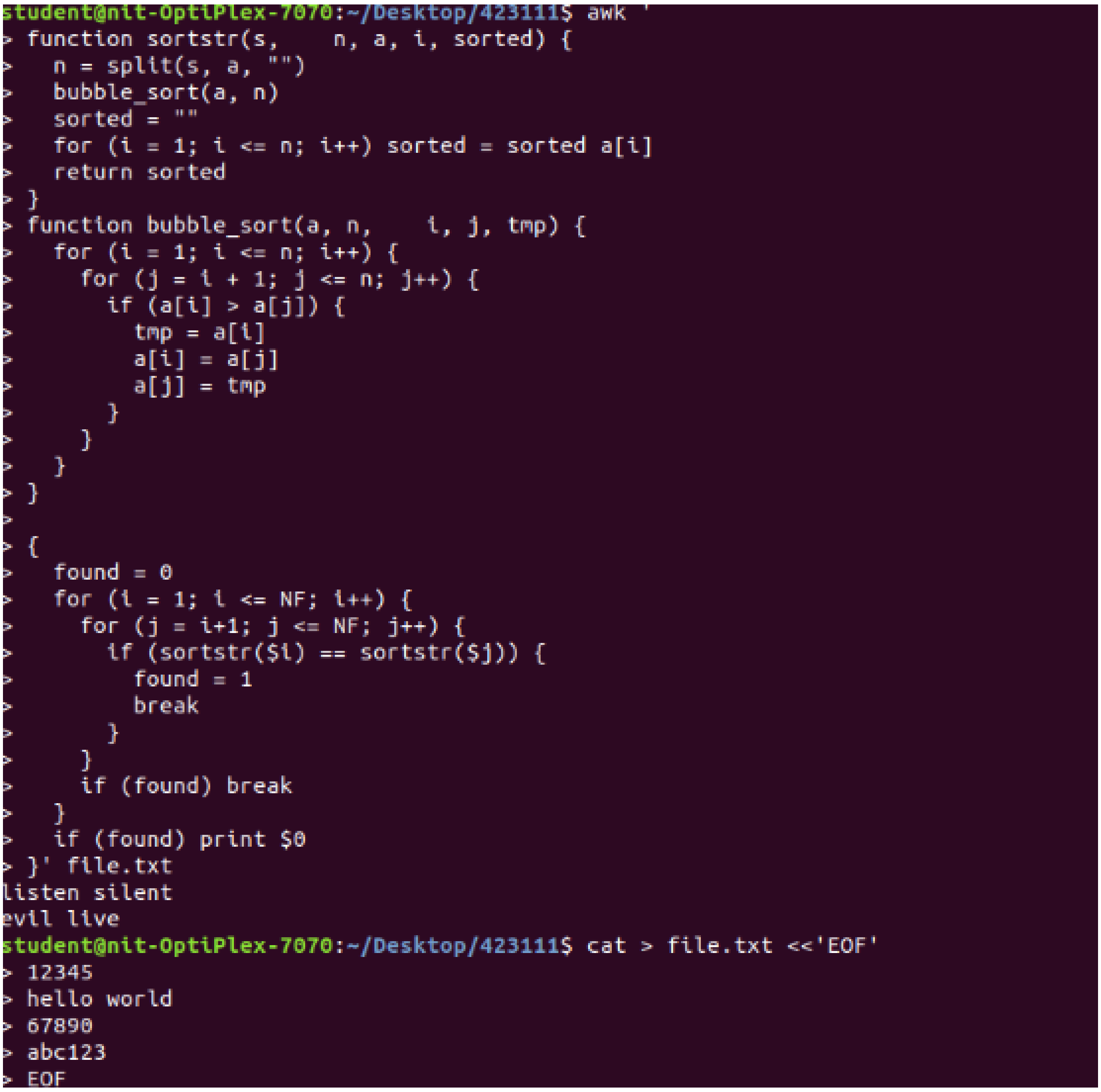
```
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '
> function sortstr(s,    n, a, i, sorted) {
>   n = split(s, a, "")
>   asort(a)
>   sorted = ""
>   for (i = 1; i <= n; i++) sorted = sorted a[i]
>   return sorted
> }
> {
>   found = 0
>   for (i = 1; i <= NF; i++) {
>     for (j = i+1; j <= NF; j++) {
>       if (sortstr($i) == sortstr($j)) { found = 1; break }
>     }
>     if (found) break
>   }
>   if (found) print $0
> }' file.txt
awk: line 19: function asort never defined
```

Unix Lab 5
roll no 423111

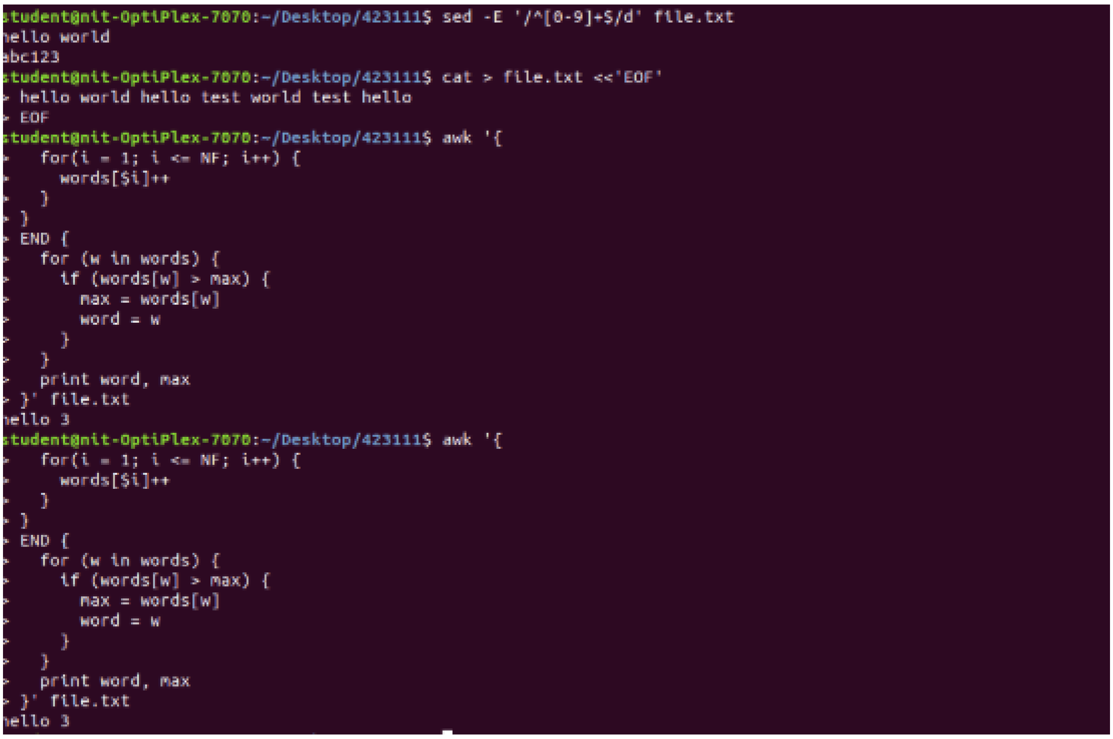


```
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '
> function sortstr(s,    n, a, i, sorted) {
>   n = split(s, a, "")
>   bubble_sort(a, n)
>   sorted = ""
>   for (i = 1; i <= n; i++) sorted = sorted a[i]
>   return sorted
> }
> function bubble_sort(a, n,    i, j, tmp) {
>   for (i = 1; i <= n; i++) {
>     for (j = i + 1; j <= n; j++) {
>       if (a[i] > a[j]) {
>         tmp = a[i]
>         a[i] = a[j]
>         a[j] = tmp
>       }
>     }
>   }
> }
>
> {
>   found = 0
>   for (i = 1; i <= NF; i++) {
>     for (j = i+1; j <= NF; j++) {
>       if (sortstr($i) == sortstr($j)) {
>         found = 1
```

Unix Lab 5
roll no 423111



```
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '
> function sortstr(s,    n, a, i, sorted) {
>   n = split(s, a, "")
>   bubble_sort(a, n)
>   sorted = ""
>   for (i = 1; i <= n; i++) sorted = sorted a[i]
>   return sorted
> }
> function bubble_sort(a, n,    i, j, tmp) {
>   for (i = 1; i <= n; i++) {
>     for (j = i + 1; j <= n; j++) {
>       if (a[i] > a[j]) {
>         tmp = a[i]
>         a[i] = a[j]
>         a[j] = tmp
>       }
>     }
>   }
> }
>
> {
>   found = 0
>   for (i = 1; i <= NF; i++) {
>     for (j = i+1; j <= NF; j++) {
>       if (sortstr($i) == sortstr($j)) {
>         found = 1
>         break
>       }
>     }
>     if (found) break
>   }
>   if (found) print $0
> }' file.txt
listen silent
evil live
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<'EOF'
> 12345
> hello world
> 67890
> abc123
> EOF
```



```
student@nit-OptiPlex-7070:~/Desktop/423111$ sed -E '/^[0-9]+$/d' file.txt
hello world
abc123
student@nit-OptiPlex-7070:~/Desktop/423111$ cat > file.txt <<'EOF'
> hello world hello test world test hello
> EOF
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{
>   for(i = 1; i <= NF; i++) {
>     words[$i]++
>   }
> }
> END {
>   for (w in words) {
>     if (words[w] > max) {
>       max = words[w]
>       word = w
>     }
>   }
>   print word, max
> }' file.txt
hello 3
student@nit-OptiPlex-7070:~/Desktop/423111$ awk '{
>   for(i = 1; i <= NF; i++) {
>     words[$i]++
>   }
> }
> END {
>   for (w in words) {
>     if (words[w] > max) {
>       max = words[w]
>       word = w
>     }
>   }
>   print word, max
> }' file.txt
hello 3
```