1. **Ans**

# Initialize variables

BEGIN {

FS = ","; # Set field separator

OFS = ","; # Set output field separator

sum\_unix = 0; # Initialize sum for Unix

sum\_dsa = 0; # Initialize sum for DSA

}

# Process each line

{

emp\_id = $1; # Extract EmpID

name = $2; # Extract Name

subject = $3; # Extract Subject

marks = $4; # Extract Marks

# Calculate average for each subject

if (subject == "Unix") {

sum\_unix += marks; # Accumulate marks for Unix

} else if (subject == "DSA") {

sum\_dsa += marks; # Accumulate marks for DSA

}

# Store details for each subject

subjects[subject][emp\_id] = { "name": name, "marks": marks };

}

# END block - calculate averages and print top 2 scorers

END {

# Calculate averages

avg\_unix = sum\_unix / NR;

avg\_dsa = sum\_dsa / NR;

# Print average scores

print "Average score in Unix:", avg\_unix;

print "Average score in DSA:", avg\_dsa;

# Print top 2 scorers in Unix

print "\nTop 2 scorers in Unix:";

print "EmpID\tName\t\tMarks";

print "-------------------------";

print\_top\_scorers(subjects["Unix"]);

# Print top 2 scorers in DSA

print "\nTop 2 scorers in DSA:";

print "EmpID\tName\t\tMarks";

print "-------------------------";

print\_top\_scorers(subjects["DSA"]);

}

# Function to print top 2 scorers

function print\_top\_scorers(subject\_data, emp\_id, details) {

PROCINFO["sorted\_in"] = "@val\_num\_desc"; # Sort in descending order

# Iterate through sorted keys

for (emp\_id in subject\_data) {

details = subject\_data[emp\_id];

print emp\_id "\t" details["name"] "\t\t" details["marks"];

if (++count == 2) break; # Print only top 2

}}

**2)Ans**

#!/bin/bash

# Company name

company\_name="Your Company Name"

# Current date

current\_date=$(date +"%Y-%m-%d %H:%M:%S")

# Header

echo "Company: $company\_name"

echo "Date: $current\_date"

echo "----------------------------------------"

echo "Username  Logged-in-time   Terminal"

echo "----------------------------------------"

# Body - List of logged-in users

who --ips --sort=-u | awk '{printf "%-11s %-16s %s\n", $1, $3, $2}' | sort -k2,2

# Footer - Total users

echo "----------------------------------------"

echo "Total users logged in: $(who -q | awk '/# users/ {print $1}')"

**3)Ans**

#!/bin/bash

# Input data file  
data\_file="emp2.txt"

# Output header for UNIX subject  
echo "Result for UNIX Subject"  
echo "---------------------------"  
echo "EmpName:ObtMark:MarksOutof50"  
echo "---------------------------"  
awk -F: '$3=="UNIX" {printf "%-15s %-8s %-15s\n", $1, $4, $5}' "$data\_file"  
echo "---------------------------"

# Total marks for each subject  
total\_marks\_dsa=50  
total\_marks\_rdbms=50  
total\_marks\_unix=50

# Output header for all subjects  
echo "Result for All Subjects"  
echo "--------------------------------------------------------------------------------------------------------------------------------------------------------------------------"  
echo "EmpName:ObtMark\_DSA:MarksOutof$total\_marks\_dsa:PercentageResult\_DSA:ObtMark\_RDBMS:MarksOutof$total\_marks\_rdbms:PercentageResult\_RDBMS:ObtMark\_UNIX:MarksOutof$total\_marks\_unix:PercentageResult\_UNIX"  
echo "--------------------------------------------------------------------------------------------------------------------------------------------------------------------------"

# Process the data file for all subjects  
awk -F: '{  
    if ($3=="DSA" || $3=="RDBMS" || $3=="UNIX") {  
        obt\_marks = $4;  
        total\_marks = ($3=="DSA") ? '$total\_marks\_dsa' : ($3=="RDBMS") ? '$total\_marks\_rdbms' : '$total\_marks\_unix';  
        percentage\_result = (obt\_marks / total\_marks) \* 100;  
        printf "%-15s %-8s %-15s %.2f%%\n", $1, obt\_marks, total\_marks, percentage\_result;  
    }  
}' "$data\_file"

# Total marks for each subject  
total\_marks\_dsa=50  
total\_marks\_rdbms=50  
total\_marks\_unix=50

# Output header for summary report  
echo "Summary Report"  
echo "--------------"

# Process the data file for all subjects  
awk -F: '{  
    if ($3=="DSA" || $3=="RDBMS" || $3=="UNIX") {  
        obt\_marks = $4;  
        total\_marks = ($3=="DSA") ? '$total\_marks\_dsa' : ($3=="RDBMS") ? '$total\_marks\_rdbms' : '$total\_marks\_unix';  
        percentage\_result = (obt\_marks / total\_marks) \* 100;

        # Calculate the total score  
        total\_score[$1] += obt\_marks;

        # Count participants  
        total\_participants++;

        # Count passed participants  
        if (percentage\_result >= 35) {  
            passed\_participants++;  
        }  
    }  
}

# Output summary information  
END {  
    print "Total Participants Appeared:", total\_participants  
    print "Total Participants Passed:", passed\_participants  
    print "---------------------------"  
    print "Participants Ranked Ist, IInd, IIIrd with Total Score:"

    # Calculate rank without using asorti  
    for (i = 1; i <= 3; i++) {  
        max\_score = -1;  
        max\_name = "";  
        for (name in total\_score) {  
            if (total\_score[name] > max\_score) {  
                max\_score = total\_score[name];  
                max\_name = name;  
            }  
        }  
        if (max\_name != "") {  
            print i ":", max\_name, max\_score;  
            delete total\_score[max\_name];  
        }  
    }  
}' "$data\_file"