**Musa Haji  
230701391  
Ex. No. 4a:**

**Employee Average Pay**Aim:

To find out the average pay of all employees whose salary is more than 6000 and number of days worked is more than 4.

Algorithm:

Create a flat file emp.dat for employees with their name, salary per day, and number of days worked, then save it.

Create an awk script emp.awk.

For each employee record, do the following:

a. If salary is greater than 6000 and the number of days worked is more than 4, then print the name and salary earned.

b. Compute total pay of the employee.

Print the total number of employees satisfying the criteria and their average pay.

Program Code:

BEGIN {

total\_pay = 0

count = 0

print "EMPLOYEES DETAILS"

}

{

salary = $2 \* $3

if ($2 > 6000 && $3 > 4) {

print $1, salary

total\_pay += salary

count++

}

}

END {

if (count > 0) {

print "no of employees are=" count

print "total pay=" total\_pay

avg\_pay = total\_pay / count

print "average pay=" avg\_pay

} else {

print "No employees meet the criteria."

}

}

Sample Input:

emp.dat – Col1 is name, Col2 is Salary Per Day, and Col3 is number of days worked.

JOE 8000 5

RAM 6000 5

TIM 5000 6

BEN 7000 7

AMY 6500 6

Sample Output:

Run the program using the below commands:

[student@localhost ~]$ vi emp.dat

[student@localhost ~]$ vi emp.awk

[student@localhost ~]$ gawk -f emp.awk emp.dat

EMPLOYEES DETAILS

JOE 40000

BEN 49000

AMY 39000

no of employees are= 3

total pay= 128000

average pay= 42666.7

Result:

Thus, the shell script correctly computes the total pay and the average pay for employees who meet the specified criteria.

**Ex No. 4b:**

**Result of Examination**

Aim:

To print the pass/fail status of a student in a class.

Algorithm:

Read the data from the file.

Get the data from each column.

Compare the marks in each subject:

a. If the marks in any subject are less than 45, then print Fail.

b. Otherwise, print Pass.

Program Code:

BEGIN {

print "NAME SUB-1 SUB-2 SUB-3 SUB-4 SUB-5 SUB-6 STATUS"

print "\_"

}

{

fail = 0

for (i = 2; i <= 7; i++) {

if ($i < 45) {

fail = 1

break

}

}

if (fail) {

status = "FAIL"

} else {

status = "PASS"

}

print $1, $2, $3, $4, $5, $6, $7, status

}

END {

print "\_"

}

Sample Input:

marks.dat – Col1 is name, Col2 to Col7 are marks in various subjects.

BEN 40 55 66 77 55 77

TOM 60 67 84 92 90 60

RAM 90 95 84 87 56 70

JIM 60 70 65 78 90 87

Sample Output:

Run the program using the below command:

[root@localhost student]# gawk -f marks.awk marks.dat

NAME SUB-1 SUB-2 SUB-3 SUB-4 SUB-5 SUB-6 STATUS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BEN 40 55 66 77 55 77 FAIL

TOM 60 67 84 92 90 60 PASS

RAM 90 95 84 87 56 70 PASS

JIM 60 70 65 78 90 87 PASS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Result:

The program correctly prints the pass/fail status for each student based on their marks in all subjects.