Ex. No. 4 a

EMPLOYEE AVERAGE PAY

Aim:

To find out the average pay of all employees whose salary is more than 6000 and no. of days worked is more than 4.

Algorithm:

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

2. Create an awk script emp.awk

3. For each employee record do

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

b. Compute total pay of employee

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

Program Code:

```
BEGIN {
    print "Employee Name | Salary"
    print "---------------------------------------"
    total_pay = 0
    count = 0
}

{

    if ($2 > 6000 && $3 > 4) {
        print $1, "|", $2*$3
        total_pay += $2*$3
        count++
    }
}

END {
    if (count > 0) {
        avg_pay = total_pay / count
        print "---------------------------------------"
        print "Total number of employees satisfying the criteria:", count
        print "Total pay of these employees:", total_pay
        print "Average pay of these employees:", avg_pay
    } else {
        print "No employees satisfying the criteria."
    }
}
```

OUTPUT-

```
[csel6@localhost ~]$ gawk -f emp.awk emp.dat
Employee Name | Salary
--------------------------------------
JOE | 40000
BEN | 49000
AMY | 39000
--------------------------------------
Total number of employees satisfying the criteria: 3
Total pay of these employees: 128000
Average pay of these employees: 42666.7
```

Ex. No. 4 b

RESULTS OF EXAMINATION

Aim:

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

Algorithm:

1. Read the data from file

2. Get a data from each column

3. Compare the all subject marks column

a. If marks less than 45 then print Fail

b. else print Pass

Program Code:

```
BEGIN {
    print "NAME SUB-1 SUB-2 SUB-3 SUB-4 SUB-5 SUB-6 STATUS"
    print "_____"
}

{

    name = $1
    marks1 = $2
    marks2 = $3
    marks3 = $4
    marks4 = $5
    marks5 = $6
    marks6 = $7


    status = "PASS"  # assume pass initially
    if (marks1 < 45 || marks2 < 45 || marks3 < 45 || marks4 < 45 || marks5 < 45 || marks6 < 45) {
        status = "FAIL"
    }


    print name, marks1, marks2, marks3, marks4, marks5, marks6, status
}

END {
    print "_____"
}
```

Output:

```
NAME SUB-1 SUB-2 SUB-3 SUB-4 SUB-5 SUB-6 STATUS
_____
       FAIL
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
```

```
[cse46@localhost ~]$ gcc round_robin_scheduling.c -o round_robin_scheduling
[cse46@localhost ~]$ ./round_robin_scheduling
Enter the number of processes: 2
Enter the time quantum: 6

Enter arrival time for Process 1: 0
Enter burst time for Process 1: 56

Enter arrival time for Process 2: 7
Enter burst time for Process 2: 54

Process ID      Arrival Time    Burst Time      Waiting Time    Turnaround Time
1               0               56              54              110
2               7               54              54              108

Average Waiting Time: 54.00
Average Turnaround Time: 109.00
```