

Ex. No.: 4a)**EMPLOYEE AVERAGE PAY****Date:03.02.25****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:**vi emp.dat**

```
JOE 8000 5
RAM 6000 5
TIM 5000 6
BEN 7000 7
AMY 6500 6
```

vi emp.awk

```
BEGIN {
    total_pay = 0;
    count = 0;
    print "EMPLOYEES DETAILS";
}
```

```

{
    name = $1;
    salary_per_day = $2;
    days_worked = $3;

    if (salary_per_day > 6000 && days_worked > 4) {
        salary_earned = salary_per_day * days_worked;
        print name, salary_earned;

        total_pay += salary_earned;
        count++;
    }
}

```

```

END {
    print "no of employees are -", count;
    print "total pay =", total_pay;

    if (count > 0) {
        print "average pay =", total_pay / count;
    } else {
        print "average pay = 0";
    }
}

```

Sample Input:

//emp.dat – Col1 is name, Col2 is Salary Per Day and Col3 is //no. of days worked

JOE 8000 5

RAM 6000 5

TIM 5000 6

BEN 7000 7

AMY 6500 6

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

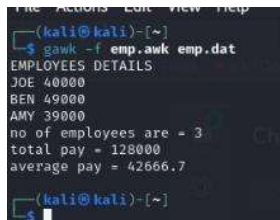
AMY 39000

no of employees are= 3

total pay= 128000

average pay= 42666.7

```
[student@localhost ~]$
```

A screenshot of a terminal window with a dark background. The prompt is (kali@kali)-[~]. The command gawk -f emp.awk emp.dat is entered and executed. The output is: EMPLOYEES DETAILS, JOE 40000, BEN 49000, AMY 39000, no of employees are = 3, total pay = 128000, and average pay = 42666.7. The prompt (kali@kali)-[~] is shown again at the bottom.

```
(kali@kali)-[~]  
$ 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  
(kali@kali)-[~]  
$
```

Result:

Thus, the program is successfully executed.

Ex. No.: 4b)

RESULTS OF EXAMINATION

Date: 03.02.25

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:

```
//marks.awk
```

vi marks.dat

```
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
```

vi marks.awk

```
BEGIN {
```

```
    print "NAME"                                STATUS";
}
```

```
{
```

```
    name = $1;
```

```
    pass = 1;
```

```
    printf "%-6s ", name;
```

```
for (i = 2; i <= 7; i++) {  
    printf "%-4d ", $i;  
    if ($i < 45) {  
        pass = 0;  
    }  
}
```

```
if (pass == 1)  
    print "PASS";  
else  
    print "FAIL";  
}
```

Input:

```
//marks.dat  
//Col1- name, Col 2 to Col7 – 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
```

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
```

```
(kali@kali)-[~]
$ gawk -f marks.awk marks.dat
NAME
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
(kali@kali)-[~]
```

Result:

Thus, the program is successfully executed.