Ex. No.: 4a)

Date: 14/12/25

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 details" }
{
    if ($2 > 6000 && $3 > 4)
    {
        print $1, "\t\t", $2 * $3
        pay = pay + $2 * $3
        count = count + 1
    }
}
END {
{
    print "no of employees are = ", count
    print "total pay = ", pay
    print "average pay = ", pay/count
}
}
```

INPUT:

jill	10000	3
bill	3000	4
will	5000	2
dill	7000	3
ben	9000	5

OUTPUT:

Employee details

ben 45000

no of employees are = 1

total pay = 45000

average pay = 45000

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 ~]$

Result:

Hence the average pay of employees whose salary is more than 6000 and no of days worked is more than 4 has been executed using AWK script. Hence the program was successfully executed and output is received.

Ex. No.: 4b)

Date: 15/2/25

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:

//marks.awk

```
BEGIN {
print "NAME", "\t", "SUB-1", "\t", "SUB-2", "\t",
      "SUB-3", "\t", "SUB-4", "\t", "SUB-5", "\t",
      "SUB-6", "\t", "STATUS"
print "_____\n"
}
{
if ( $2<45 || $3<45 || $4<45 || $5<45 || $6<45
                              || $7<45 )
{
print $1, "\t", $2, "\t", $3, "\t", $4, "\t",
      $5, "\t", $6, "\t", $7, "\t", "FAIL"
}
else
{
print $1, "\t", $2, "\t", $3, "\t", $4, "\t", $5,
      "\t", $6, "\t", $7, "\t", "PASS" }}
END{
print "_____" "\n" }
```

INPUT

Jill	50	59	64	76	50	75
Dill	78	67	73	85	95	66
Bill	70	85	64	89	65	79
Roger	90	67	55	88	80	77
Ben	20	30	40	40	30	30

OUTPUT

Name	SUB-1	SUB-2	SUB-3	SUB-4	SUB-5	SUB-6	STATUS
Jill	50	59	64	76	50	75	PASS
Dill	78	67	73	85	95	66	PASS
Bill	70	85	64	89	65	79	PASS
Roger	90	67	55	88	80	77	PASS
Ben	20	30	40	40	30	30	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

Result:

Thus the AWK script to print the examination result is been programmed and executed successfully.