Ex. No.: 4a)

Date: 14/2/25

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 "EMPLOYEES DETAILS" }
{ #salary should be greater than 6000 and days more than 4
    if ($2>6000 && $3>4)
    {
      print $1, "\t \t", $2*$3
      pay=pay+$2*$3
      count=count+1
    }
    }
  END{
    { #action part
      print "no. of employees are =", count
      print "total pay=", pay
      print "average pay=", pay/count
    }
  }
```

emp.dat - col1 is name, col2 is salary per day and col3 is no. of days

emp.dat

Sreya	7500	5
Yashv	6570	5
Teju	5050	6
Uma	7600	7
Thomsy	5500	6

Input

Sreya	7500	5
Yashv	6570	5
Teju	5050	6
Uma	7600	7
Thomsy	5500	6

Output

Employees Details

Sreya	37500
yashv	32850
Teju	53200

no. of employees are=3

total pay=123550

average pay=41183.3

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:

Thus the AWK script for employee average pay is successfully executed

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 5", "\t", "SUB- 6", "\t", "STATUS"
  print "_____\n"}

{#BODY
  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", "FAI
  }
  else
  {
  print $1, "\t", $2, "\t", $3, "\t", $4, "\t", $5, "\t", $6, "\t", $7, "\t",
  "PAS
  }
}
END{
  print "_____\n"}
```

marks.dat

```
sreya   55  67  77  88  99  84
Yash    78  58  93  79  63  83
Teju    88  98  91  71  72  81
varsha  88  68  92  62  57  86
```

Output

NAME	SUB-1	SUB-2	SUB-3	SUB-4	SUB-5	SUB-6	Status
Sreya	55	67	77	88	99	84	PASS
Yashv	78	58	93	79	63	83	PASS
Teju	88	98	91	71	72	86	PASS
Varsha	83	68	92	62	57	86	PASS

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 for results of examination is successfully executed