Exp No. 9 Date : 30-04-2019

AWK SCRIPTING

Aim :

To perform simple text processing AWK Scripting.

Programs :

Question 1 :-

Write a awk script that accepts date argument in the form of mm-dd-yy and displays it in the following format. The script should check the validity of the argument and in the case of error, display a suitable message.

Algorithm :

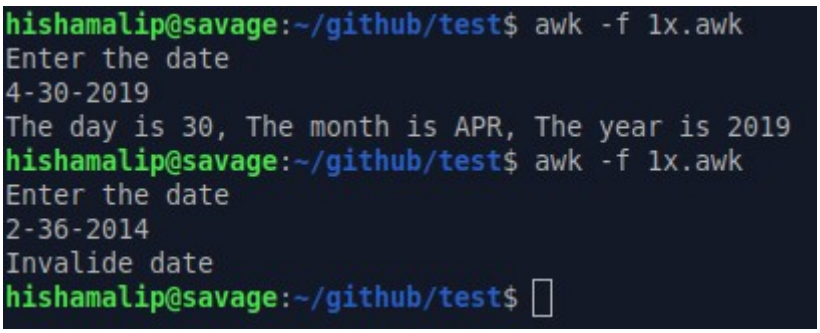
1. Start
2. Read argument
3. If invalid or null argument , print error message
4. Else store day, year, month separately
5. Replace numeric month to alphabetical month
6. Print “The date is “ day “Month is “ month “Year is “ year
7. Stop

Program :

```
BEGIN{      FS="-";
            print "Enter the date"
            getline < "/dev/tty"
                  flag=1;
if(   (((($3%4!=0)&&($1==2)&&($2>28))||(($3%4==0)&&($1==2)&&($2>29)))||
     (((($1==1)||($1==3)||($1==5)||($1==7)||($1==8)||($1==10)||($1==12)) &&($2>31))||
    ((($1==4)||($1==6)||($1==9)||($1==11)) && ($2>30) )||($1<1)||($2<1)||($3<1)||($1>12)  )
            flag=0;
            if(flag==0)
                  print "Invalide date"
            else {
                  if($1==1)
                        month="JAN";
                  else if($1==2)
                        month="FEB";
                  else if($1==3)
                        month="MAR";
                  else if($1==4)
                        month="APR";
                  else if($1==5)
```

```
                month="MAY";
        else if($1==6)
                month="JUNE";
        else if($1==7)
                month="JULY";
        else if($1==8)
                month="AUG";
        else if($1==9)
                month="SEP";
        else if($1==10)
                month="OCT";
        else if($1==11)
                month="NOV";
        else
                month="DEC";
    print "The day is "$2", The month is "month", The year is "$3
    }
}
```

Output:



Question 2 :-

Write an awk script to delete duplicated line from a text file. The order of the original lines must remain unchanged.

Algorithm :

1. Start
2. Read argument
3. If duplicate lines not found
4. Print line
5. Move index to next line
6. Stop

Program :-

```
BEGIN{ print "File after deleting duplicate lines"}
{
    if(!x[$0]++)
    print $0
}
```

Output :

Question 3 :-

Write an awk script to find out total number of books sold in each discipline as well as total book sold based on the given table.

electrical 34
mechanical 67
electrical 80
computers 43
mechanical 65
civil 198
computers 64

Algorithm :

1. Start
2. Read argument
3. Set count of all disciplines&total to 0
4. Add total no of books
4. If a disciplines is found add book count of that discpline
5. Print book sold count and total count
6. Stop

Program :

```
BEGIN{  total=0; elec=0; mech=0; comp=0; civil=0   }
{
   total=total+$2
   if($1=="electrical")
     elec=elec+$2
   if ($1=="mechanical")
      mech=mech+$2
   if($1=="computers")
      comp=comp+$2
   if ($1=="civil")
      civil=civil+$2
}
END{
      print "Electrical : ",elec
      print "Mechanical : ",mech
      print "Computer : ",comp
      print "Civil : ",civil
      print "\nTotal no. of Books : ",total
  }
```

Output:

Question 4 :-

Write an awk script to compute gross salary of an employee accordingly to rule given below :

If basic salary < 10000 then DA = 45% of the basic and HRA =15% of basic
If basic salary >= 10000 then DA =50% of the basic and HRA =20% of basic

Algorithm :

1. Start
2. Read Salary
3. If (salary < 10000) then
4. DA = 45% of salary

5. HRA = 15% of salary
6. Else
7. DA = 45% of salary
8. HRA = 50% of salary
9. End if
10. Gross salary = DA+HRA+salary
11. Print gross salary
12. Stop

Program :

```
BEGIN{  printf "Enter the Salary of employee : "
        getline salary
        if(salary<10000)
        {
            HRA=.15*salary
            DA=.45*salary
        }
        else
        {
            HRA=.2*salary
            DA=.5*salary
        }
        gross_salary=salary+HRA+DA
        printf "Gross Salary : %.2f/-\n",gross_salary
}
```

Output :

Result :

Simple text processing using AWK Scripting are familiarized and output is obtained successfully