Awk Scripting

Rwithik Manoj

April 29, 2019

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:

   (a) Start.

   (b) Check if the date is valid. Exit if its not.

   (c) If it is a valid date, find the corrosponding month. And print the output in the format specified.

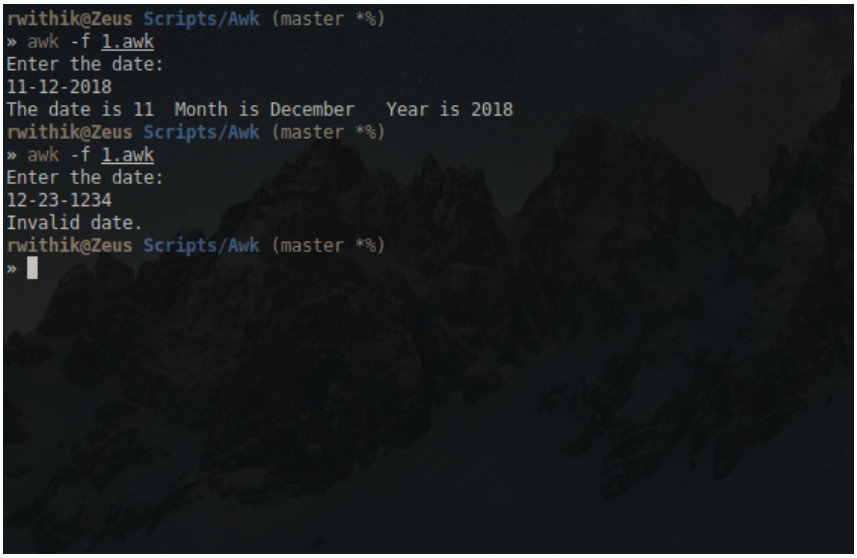
   (d) Stop.

   Script:

```
#!/usr/bin/awk
BEGIN{
FS="-"
print "Enter the date:"
getline < "/dev/tty"
flag=1;
if((($3%4!=0) && ($2==2) && ($1>28)) || (($3%4==0) && ($2==2) && ($1>29)) || $2 > 12)
    flag=0;
if(flag==0)
        print"Invalid date."
else
{
         if($2==1)
                temp="January";
         else if($2==2)
                temp="February";
         else if($2==3)
                temp="March";
         else if($2==4)
                temp="April";
         else if($2==5)
                temp="May";
         else if($2==6)
                temp="June";
         else if($2==7)
                temp="July";
         else if($2==8)
                temp="August";
         else if($2==9)
                temp="September";
         else if($2==10)
                temp="October";
         else if($2==11)
                temp="November";
         else
```

```
                    temp="December";
print "The date is " $1 "\tMonth is " temp "\tYear is " $3
}
}
```

Output:



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

   (a) Start.

   (b) Print the line only if seen[$0] is zero, ie the count of the current line is zero.

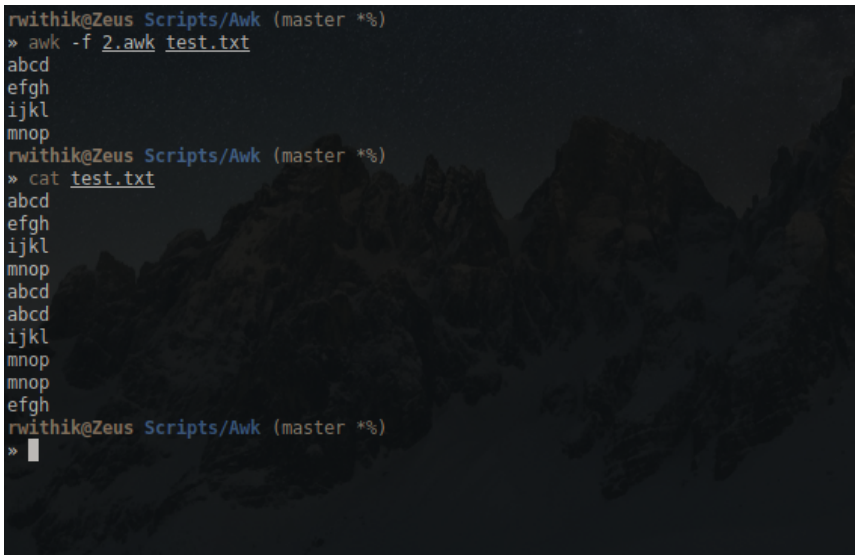
   (c) Stop.

Script:

```
#!/bin/awk

!seen[$0]++ {print $1}
```

Output:



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.
Algorithm:

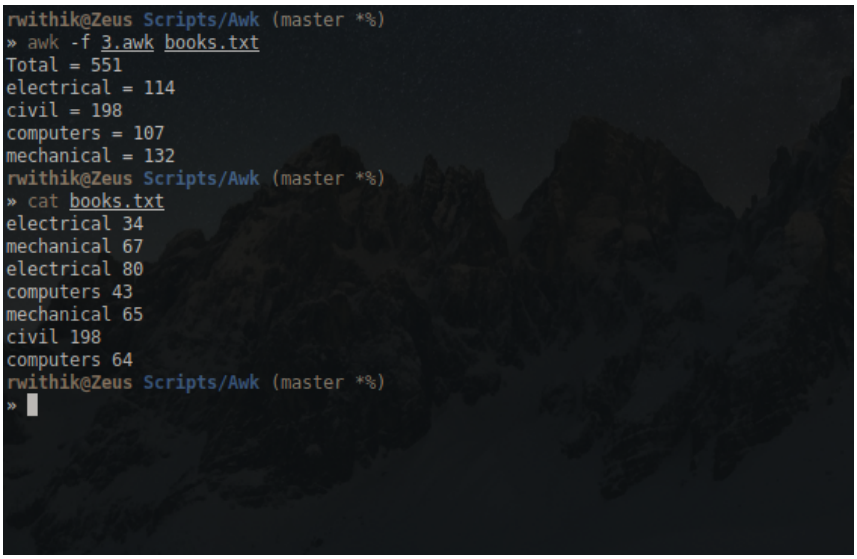
   (a) Start.
   (b) Initalise sum to 0.
   (c) Add the second field to the sum.
   (d) Add the Second field to the corrosponding array element.
   (e) Print the values.
   (f) Stop.

Script:

```
#!/bin/awk

BEGIN{
    sum = 0
}
{
    sum = sum + $2;
    subject[$1] = subject[$1] + $2;
    }
END{
    printf "Total = " sum "\n"
    for(word in subject) printf word " = " subject[word] "\n"
}
```

Output:



4. Write an awk script to compute gross salary of an employee accordingly to rule given below: If basic salary is less than 10000, then DA = 45% of the basic and HRA = 15% of basic. If basic salary is greater than 10000, then DA = 50% of the basic and HRA = 20% of basic.
Algorithm:

   (a) Start.

   (b) Check if the basic pay is less than 10000. If it is, calculate the total salary as BP + 0.45BP + 0.15BP.

   (c) Otherwise, the total salary is BP + 0.5BP + 0.2BP.

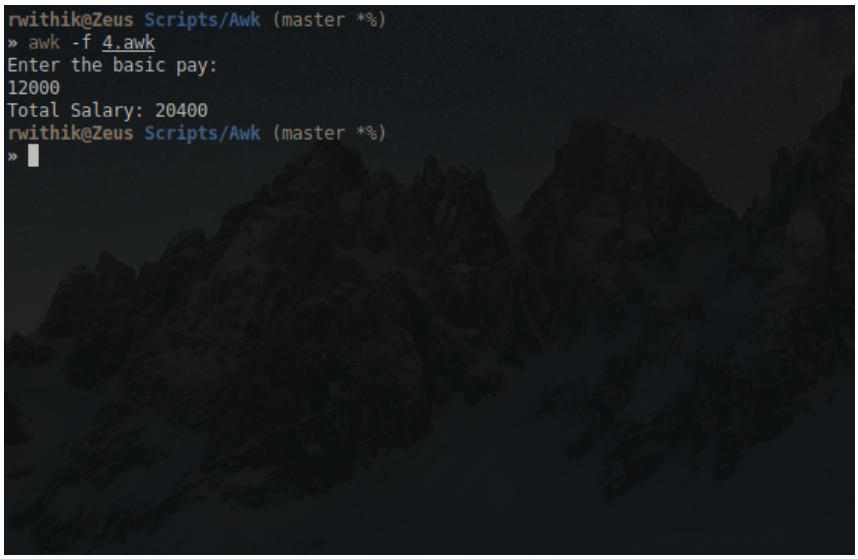
   (d) Stop.

Script:

```
#!/bin/awk

BEGIN{
    print "Enter the basic pay:"
    getline < "/dev/tty"
    totalSal = $1
    if ($1 < 10000)
        totalSal = 0.45 * $1 + 0.15 * $1 + $1
    else
        totalSal = 0.50 * $1 + 0.20 * $1 + $1
    print "Total Salary: " totalSal
}
```

Output:


rwithik@Zeus Scripts/Awk (master *%)
» awk -f 4.awk
Enter the basic pay:
12000
Total Salary: 20400
rwithik@Zeus Scripts/Awk (master *%)
»