

FOSS LAB REPORT

Neethu S

Roll no: 42

TVE18CS043

College of Engineering Trivandrum

7 Feb 2020

1 Experiment 11

1.1 (a) Perl Scripting

1.1.1 Aim

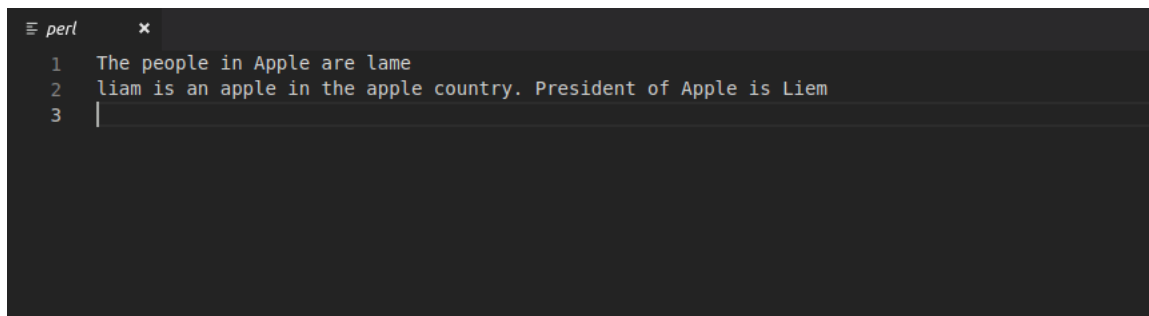
Create a text file and answer the following queries :

- Search for the pattern 'apple' in the file and display the number of occurrences.
- Count the number of words that ends with 'e'
- Count the number of words that starts with 'ap'
- Search for words containing 'a' or 's'
- Search for words containing zero or more occurrence of 'e'
- Search for words containing one or more occurrence of 'e'
- Search for words containing the letters 'l' and 'm', with any number of characters in between

1.1.2 Overview

Perl script

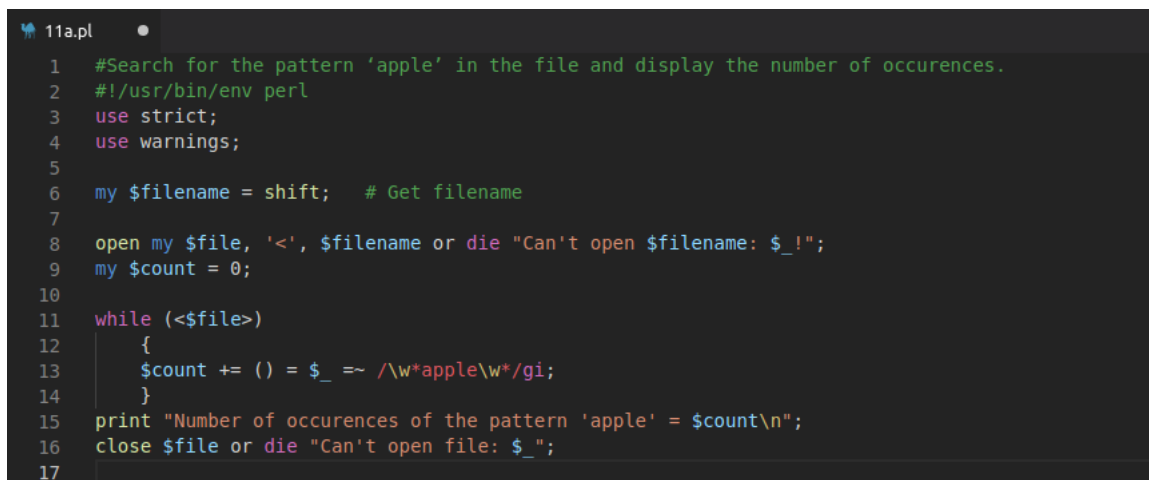
The text file given as input is:



```
perl x
1 The people in Apple are lame
2 liam is an apple in the apple country. President of Apple is Liam
3 |
```

- Search for the pattern 'apple' in the file and display the number of occurrences.

The code is as follows:



```
11a.pl
1 #Search for the pattern 'apple' in the file and display the number of occurrences.
2 #!/usr/bin/env perl
3 use strict;
4 use warnings;
5
6 my $filename = shift; # Get filename
7
8 open my $file, '<', $filename or die "Can't open $filename: $_!";
9 my $count = 0;
10
11 while (<$file>)
12 {
13     $count += () = $_ =~ /\w*apple\w*/gi;
14 }
15 print "Number of occurrences of the pattern 'apple' = $count\n";
16 close $file or die "Can't open file: $_";
17
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11a.pl perl
Number of occurrences of the pattern 'apple' = 4
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

b) Count the number of words that ends with 'e'

The code is as follows:

```
11b.pl
1  #Count the number of words that ends with 'e'
2  #!/usr/bin/env perl
3  use strict;
4  use warnings;
5
6  my $filename = shift; # Get filename
7
8  open my $file, '<', $filename or die "Can't open $filename: $_!";
9  my $count = 0;
10
11 while (<$file>)
12 {
13     $count += () = $_ =~ /\w*e\b/gi;
14 }
15 print "Number of words that end with 'e' = $count\n";
16 close $file or die "Can't open file: $_";
17
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11b.pl perl
Number of words that end with 'e' = 9
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

c) Count the number of words that starts with 'ap'

The code is as follows:

```
11c.pl
1  #Count the number of words that begins with 'ap'
2  #!/usr/bin/env perl
3  use strict;
4  use warnings;
5
6  my $filename = shift; # Get filename
7
8  open my $file, '<', $filename or die "Can't open $filename: $_!";
9  my $count = 0;
10
11 while (<$file>)
12 {
13     $count += () = $_ =~ /\bap\w*/gi;
14 }
15 print "Number of words that begins with 'ap' = $count\n";
16 close $file or die "Can't open file: $_";
17
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11c.pl perl
Number of words that begins with 'ap' = 4
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

d) Search for words containing 'a' or 's'

The code is as follows:

```
11d.pl
1  #Search for words containing 'a' or 's'
2  #!/usr/bin/env perl
3  use strict;
4  use warnings;
5
6  my $filename = shift; # Get filename
7  open my $file, '<', $filename or die "Can't open $filename: $_!";
8
9  print "Words containing 'a' or 's' are:\n";
10
11 while (my $line = <$file>){
12     chomp $line;
13     foreach my $string (split /\s+/, $line ){
14         if($string =~ /\w*(a|s)\w*/gi )
15             print "$string\n";
16     }
17 }
18 close $file or die "Can't open file: $_";
19
20
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11d.pl perl
Words containing 'a' or 's' are:
Apple
are
lame
liam
is
an
apple
apple
President
Apple
is
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

e) Search for words containing zero or more occurrence of 'e'

The code is as follows:

```
11e.pl
1  #Search for words containing zero or more occurrence of 'e'
2  #!/usr/bin/env perl
3  use strict;
4  use warnings;
5
6  my $filename = shift; # Get filename
7  open my $file, '<', $filename or die "Can't open $filename: $_!";
8
9  print "Words containing zero or more occurrences of 'e' are:\n";
10 my $count = 0;
11 while (my $line = <$file>) {
12     chomp $line;
13     foreach my $string (split /\s+/, $line) {
14         if($string =~ /e*/) {
15             $count++;
16             print "$string\n";
17         }
18     }
19 }
20 print "Count = $count \n";
21 close $file or die "Can't open file: $_";
22
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11e.pl perl
Words containing zero or more occurrences of 'e' are:
The
people
in
Apple
are
lane
liam
is
an
apple
in
the
apple
country.
President
of
Apple
is
liem
Count = 19
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

f) Search for words containing one or more occurrence of 'e'

The code is as follows:

```
11f.pl
1  #Search for words containing one or more occurrence of 'e'
2  #!/usr/bin/env perl
3  use strict;
4  use warnings;
5
6  my $filename = shift; # Get filename
7  open my $file, '<', $filename or die "Can't open $filename: $_!";
8
9  print "Words containing one or more occurrences of 'e' are:\n";
10 my $count = 0;
11 while (my $line = <$file>) {
12     chomp $line;
13     foreach my $string (split /\s+/, $line) {
14         if($string =~ /e/) {
15             $count++;
16             print "$string\n";
17         }
18     }
19 }
20 print "Count = $count \n";
21 close $file or die "Can't open file: $_";
22
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11f.pl perl
Words containing one or more occurrences of 'e' are:
The
people
Apple
are
lame
apple
the
apple
President
Apple
Liem
Count = 11
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

g) Search for words containing the letters 'l' and 'm', with any number of characters in between

The code is as follows:

```
11g.pl
1  #Search for words containing the letters 'l' and 'm', with any number of characters in between
2  #!/usr/bin/env perl
3  use strict;
4  use warnings;
5
6  my $filename = shift; # Get filename
7  open my $file, '<', $filename or die "Can't open $filename: $_!";
8
9  print "Words containing the letters 'l' and 'm', with any number of characters in between:\n";
10 my $count = 0;
11 while (my $line = <$file>) {
12     chomp $line;
13     foreach my $string (split /\s+/, $line) {
14         if($string =~ /\bl..m\b/gi){
15             $count+=1;
16             print "$string\n";
17         }
18     }
19 }
20 print "Count = $count \n";
21 close $file or die "Can't open file: $_";
22
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ perl 11g.pl perl
Words containing the letters 'l' and 'm', with any number of characters in between:
liam
liem
Count = 2
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

1.1.3 Result

The simple text processing was done successfully using *perl* script and the output was verified. The perl script was run on Ubuntu 18.04.

1.2 (b) Awk Scripting

1.2.1 Aim

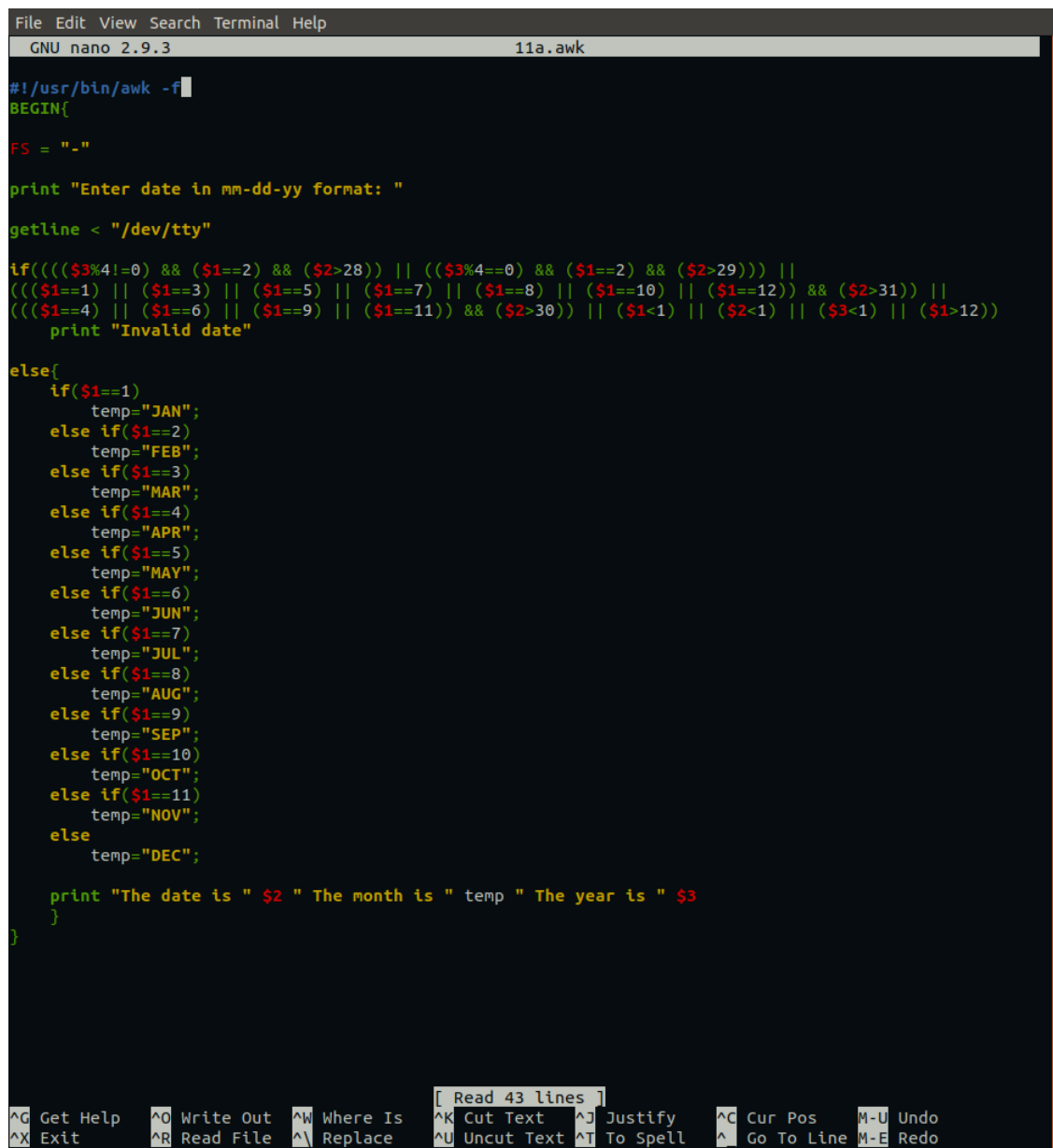
To be able to write the awk script.

1.2.2 Overview

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.

- Sample I/p : 12-10-2008
- Sample O/p : The day is 10 The month is OCT The year is 2008

The awk script is as follows:



```
File Edit View Search Terminal Help
GNU nano 2.9.3 11a.awk

#!/usr/bin/awk -f
BEGIN{
    FS = "-"

    print "Enter date in mm-dd-yy format: "

    getline < "/dev/tty"

    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))
        print "Invalid date"
    else{
        if($1==1)
            temp="JAN";
        else if($1==2)
            temp="FEB";
        else if($1==3)
            temp="MAR";
        else if($1==4)
            temp="APR";
        else if($1==5)
            temp="MAY";
        else if($1==6)
            temp="JUN";
        else if($1==7)
            temp="JUL";
        else if($1==8)
            temp="AUG";
        else if($1==9)
            temp="SEP";
        else if($1==10)
            temp="OCT";
        else if($1==11)
            temp="NOV";
        else
            temp="DEC";

        print "The date is " $2 " The month is " temp " The year is " $3
    }
}
```


Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11a.awk
Enter date in mm-dd-yy format:
12-04-2001
The date is 04 The month is DEC The year is 2001
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11a.awk
Enter date in mm-dd-yy format:
20-02-2000
Invalid date
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11a.awk
Enter date in mm-dd-yy format:
02-30-2119
Invalid date
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11a.awk
Enter date in mm-dd-yy format:
04-28-2020
The date is 28 The month is APR The year is 2020
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

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

The text file given as input is:

```
11b.awk  text x
1 blue
2 blue
3 green
4 black
5 black
6 yellow
7 pink
8 pink
9 pink
10
```

The awk script is as follows:

```
#!/bin/awk -f
BEGIN
{
    FS="\n"
}
{
    if(var!=$1 && NR>=0)
        print line
    var=$1
    line=$0
}
END
{
    print line
}
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11b.awk text
blue
green
black
yellow
pink
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

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

The text file given as input is:

```
text1
1 electrical 34
2 mechanical 67
3 electrical 80
4 computers 43
5 mechanical 65
6 civil 198
7 computers 64
8
```

The awk script is as follows:

```
#!/bin/awk -f
{
    arr[$1]+=$2;
    total+=$2;
}
END{
    for (i in arr){
        print i " " arr[i];
    }
    print "Total " total
}
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11c.awk text1
civil 198
computers 107blue
electrical 114blue
mechanical 132green
Total 551black
neethu@neethu-Inspiron-15-3567:~/s4/perl$
```

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.

The awk script is as follows:

```
#!/usr/bin/awk -f

BEGIN{

    print "Enter the Basic Salary: ";

    getline BS < "/dev/tty";

    if(BS < 10000){

        DA = 45/100 * BS;

        HRA = 15/100 * BS;

    }

    else{

        DA = 50/100 * BS;

        HRA = 20/100 * BS;

    }

    GS = BS + DA + HRA;

    print "Gross salary = " GS

}
```

Sample input and output:

```
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11d.awk
Enter the Basic Salary:
12000      3      green
Gross salary = 20400
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11d.awk
Enter the Basic Salary:
10000      7      pink
Gross salary = 17000
neethu@neethu-Inspiron-15-3567:~/s4/perl$ awk -f 11d.awk
Enter the Basic Salary:
9897
Gross salary = 15835.2
```

1.2.3 Result

The simple text processing was done successfully using *awk* script and the output was verified. The *awk* script was run on Ubuntu 18.04.
