Experiment-7

Aim: write Ruby program reads a number and calculates the factorial value of it and prints the same

program

```
puts "Enter a number>>"
n= gets.to_i
f=1
 i=1
 while i<=n do
  f=f*i
  i=i+1
  end
  puts "factorial of #{n} is #{f}"
```

output

```
$ ruby 7.rb
 Enter a number
 4
  factorial of 4 is 120
```

Experiment No: 8

8.

Aim: write a Ruby program which counts number of lines in a text files using its regular Expressions facility

Program

```
Print "Enter file name:";
fname=gets.chomp;
count = IO.readlines(fname).size
Puts "There are #{count} lines in #{fname}";
```

output

```
$ ruby regEx.rb
    Enter file regEx.rb
    There are 5 lines in regEx.rb
```

Scanned by TapScanner

Experiment No: 9

AIm: write a Ruby Program the uses Iterator to find out the length of a string

Program

```
Print "Enter a string:"
str = gets.chomp
 c=0
 str.each_char do |i|
 c=c+1
 end
  puts "Length of #{str}:#{c}"
```

output:

```
$ ruby 9.rb
  Enter a string: hello vyshu
  length of hello vyshu:11
```

Experiment - 10

Aim: write simple Ruby Programs that uses arrays in Ruby

Program

```
Puts "Bubble Sort"
Puts "= = = = = = ="
Prints "Enter the size of the array:"
n=gets.to_i
a=Array.new(n)
Puts "Enter the elements:"
for i in 0..n-1
a[i]=gets.to_i
end
#Bubble sort Algorithm
t=0
for i in 0..n-1
for j in i..n-1
if (a[i]>a[j])
   t=a[j]
   a[j]=a[i]
    a[i]=t
    end
   end
   end
Puts "After sorting:"
for i in 0..n-1
 Print "#{a[i]}"
end
```

output {

```
Bubble Sort
======
Enter the size of array: 3
Enter the elements:
33
15
24
After sorting:
15 24 33
```

Experiment – 11

AIM: write Programs which uses associative arrays Concept of Ruby.

Program

```
Creators = Hash.new;
Creators = { "java" => "vvit", "webtechnologies" => "cse",
                    "c language" => "lit" };

 size = creators.length;
 puts "size of hash is: #{size}";
 creators.each do |key, val|
 puts "#{key} invented by #{val}"
 end
```

output

```
$ ruby 11.rb
   size of hash is: 3
   java invented by vvit
   Webtechnologies invented by cse
   c language invented by it
```

Experiment - 12

Aim: Write a Ruby Program which uses Math module to find area of a triangle

Program

```
Print "Enter a,b,c values:"
a=gets.to_i
b=gets.to_i
c=gets.to_i
s=(a+b+c)/2
area=Math.sqrt(s*(s-a)*(s-b)*(s-c)).round(2)
Print "Area of Triangle #{area}"
```

Output

```
$ ruby 12.rb
  Enter a,b,c values:
  3
  4
  5
  Area of Triangle = 6.0
```

Scanned by TapScanner

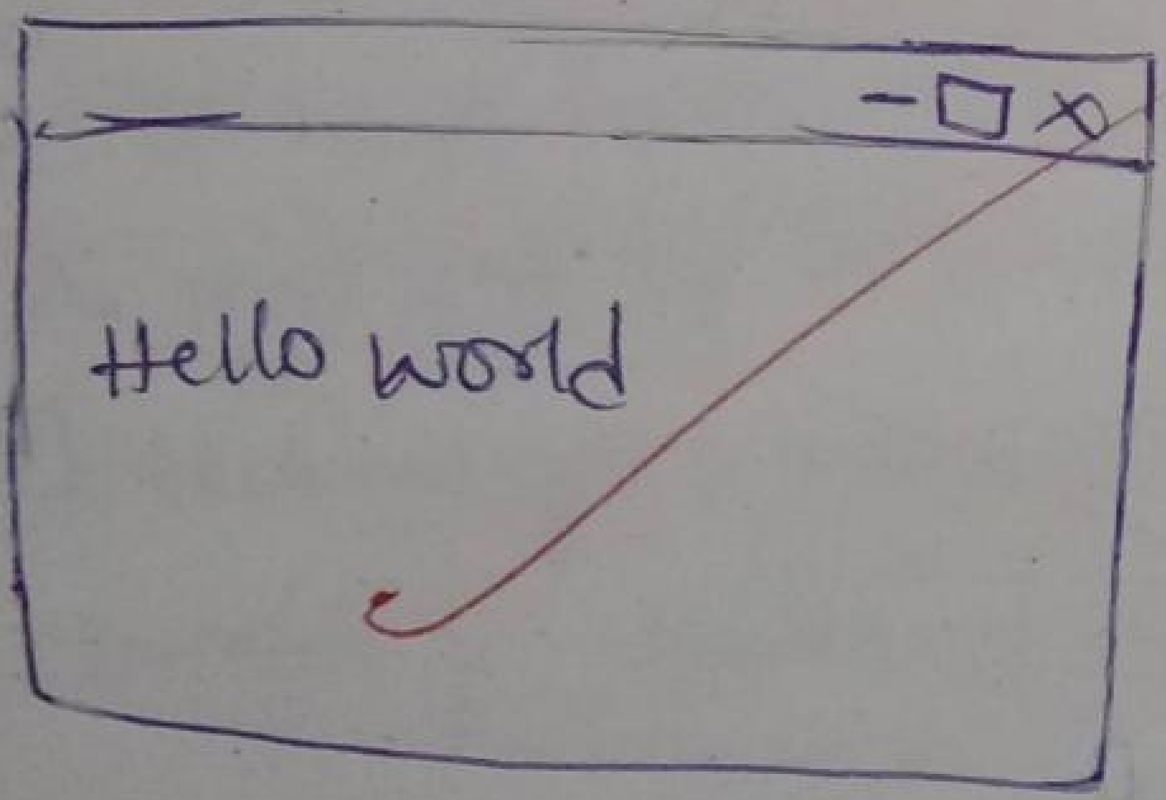
Experiment -13

AIM: write Ruby program which uses tk module to display a window

Program

```
require 'tk'
root = TKRoot.new {title "GUI window"}
TKLabel.new(root) do
  text 'Helloworld!'
  Pack("side"=>"right", "Padx"=>"100", "Pady"=>"100")
  end
  TK.mainloop
```

output



Scanned by TapScanner

Experiment No: 15

AIM: write a program which illustrates the use of associative array in Perl

Program

```
%ages=('kiran'=>19,'vijay'=>21,'raju'=>20);
Print "Original Array:\n";
 Print " ==========\n";
 while (($key)= each %ages)
 {
   Print "$key is $ages{$key} years old\n";
 }
 $ages{'mayur'}=24;
 Print "\nAfter additing element:\n";
 Print "==========\n";
 while(($key)=each %ages)
 {
   Print "$key is $ages{$key} years old\n";
 }
 delete($ages{'vijay'});
 Print "\nAfter deleting element:\n";
 Print "==========\n";
 @all_keys=keys(%ages);
 Print "keys are: @all_keys\n";
 @all_values =values (%ages);
 Print "values are :@all_values";
```

Output

```
$ Perl 15.pl
 Orignal Array:
 ========
 kiran is 19 years old
  raju is 20 years old
  vijay  is 21 years old
 After adding element:
 ========
 vijay is 21 years old
 mayr is 24 years old
 kiran is 19 years old
 raju is 20 years old

After deleting element:
==========
Keys are: mayur kiran raju
values are: 24 19 20
```

Experiment-16

AIM: write perl program takes set names along the command line and prints whether they are regular files or special files

Program

```
$len=@ARGV;
for($i=0;$i<$len;$i++)
{
    if(-e $ARGV[$i])
    {
     if(-T $ARGV[$i])
     {
       Print "$ARGV[$i] is a text file\n";
     }
     else
     {
       Print "$ARGV[$i] is a special file\n";
     }
    }
    else
    {
     Print "$ARGV[$i] does not exists";
    }
}
```

output:

```
$perl 16.pl 45.pl
45.pl does not exists
$perl 16.pl 7.rb
7.rb is a text file
```

Experiment No: 17

AIM: write a perl program to implement UNIX 'Passwd' Program

Program

```
my $salt ='';
my $encrypted = '';
 my $password = '';
 my $use = 'Usage: please Provide password for encrypt';
 my @saltchars = ('.', '/', 0..9, 'A'..'Z', 'a'..'z');
 my $args = @ARGV;
 if( $args < 1 || $args > 2)
 {
   print "$use\n";
   exit;
 }
 $password = $ARGV[0];
 if( $args == 1)
 { }
 else
 { }
 $ salt = join('', @saltchars[rand(64), rand(64)]);
$salt = $ARGV[1];
$encrypted = crypt( $password, $salt);
Print "$Password -> $encrypted \n";
```

output:

```
$ perl -w 17.pl hellomahi qi
hellomahi -> qioEYCIoYJcCU

$ perl -w 17.pl sacet123 a1
sacet123 -> a152j1A3GRZ/.
```

[illegible]