**Department of Computer Science & Application**

**Bachelor of Computer Application (BCA)**

**Semester I**

**Course Code UGCSA104**

**Course Title WDD LAB MANUAL**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **Page** |
| **1** | Syllabus | 3 |
| **2** | Do’s and Don’ts | 4 |
| **3** | Instruction to the students | 5 |
| **4** | Lab PEO | 6 |
| **5** | LAB Plan | 12 |

**DO’S AND DONT’S**

**DO’S**

1.Student should get the record of previous experiment checked before starting the new experiment.

2.Read the manual carefully before starting the experiment.

3.Turn off the compute before leaving the lab unless a member of lab staff has specifically told you not to do so.

4.Before switching on the power supply, get the circuit connections checked.

5.Get your outputs checked by the teacher.

6.PC and Apparatus must be handled carefully.

7.Maintain strict discipline.

8.Keep your mobile phone switched off or in vibration mode.

9.Students should get the experiment allotted for next turn, before leaving the lab.

**DONT’S**

1.Don’t use internet, internet chat of any kind in your regular lab schedule.

2.Do not download or upload of MP3, JPG or MPEG files.

3.No games are allowed in the lab sessions.

4.No hardware including USB drives can be connected or disconnected in the labs without prior permission of the lab in-charge.

5.Do not leave the without permission from the teacher.

6.If you are having problems or questions, please go to either the faculty, lab in-charge or the lab supporting staff. They will help you. We need your full support and cooperation for smooth functioning of the lab.

**Instructions to the Students**

**General Instructions**

• Maintain separate observation copy for each laboratory.

• Observations or readings should be taken only in the observation copy.

• Get the readings counter signed by the faculty after the completion of the experiment.

• Maintain Index column in the observation copy and get the signature of the faculty before leaving the lab.

**Before Entering the Lab**

• The previous experiment should have been written in the practical file, without which the students will not be allowed to enter the lab.

• The students should have written the experiment in the observation copy that they are supposed to perform in the lab.

• The experiment written in the observation copy should have aim, apparatus required, circuit diagram/algorithm, blank observation table (if any), formula (if any), programmed (if any), model graph (if any) and space for result.

**When Working in the Lab**

• Necessary equipment’s/apparatus should be taken only from the lab assistant by making an issuing slip, which would contain name of the experiment, names of batch members and apparatus or components required.

• Never switch on the power supply before getting the permission from the faculty.

**Before Leaving the Lab**

• The equipment’s/components should be returned to the lab assistant in good condition after the completion of the experiment.

• The students should get the signature from the faculty in the observation copy.

• They should also check whether their file is checked, and counter signed in the index.

1. Design the following static web pages required for an online book store web site.

1) **HOME PAGE:**

The static home page must contain three **frames**.

**Top frame**: Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

**Left frame:** At least four links for navigation, which will display the catalogue of respective links.

For e.g.: When you click the link **“MCA”** the catalogue for **MCA** Books should be displayed in the Right frame. **Right frame**: The *pages to the links in the left frame must be loaded here*. Initially this page contains description of the web site.

# head.html<html>

<head>

<title>Head Page</title>

</head>

<body>

<font face="Arial Black" size="3">

<table border="1" cellspacing="2" cellpadding="5" width="100%">

<tr>

<td align="center"><img src="6.jpg" width="20" height="20"/></td>

<td colspan="4" align="center">Web Site Name</td>

</tr>

<tr>

<td align="center"><a href="description.html" target="des\_page">HOME</a></td>

<td align="center"><a href="login.html" target="des\_page">LOGIN</a></td>

<td align="center"><a href="registration.html" target="des\_page">REGISTRATION</a></td>

<td align="center"><a href="catalogue.html" target="des\_page">CATALOGUE</a></td>

<td align="center"><a href="cart.html" target="des\_page">CART</a></td>

</tr>

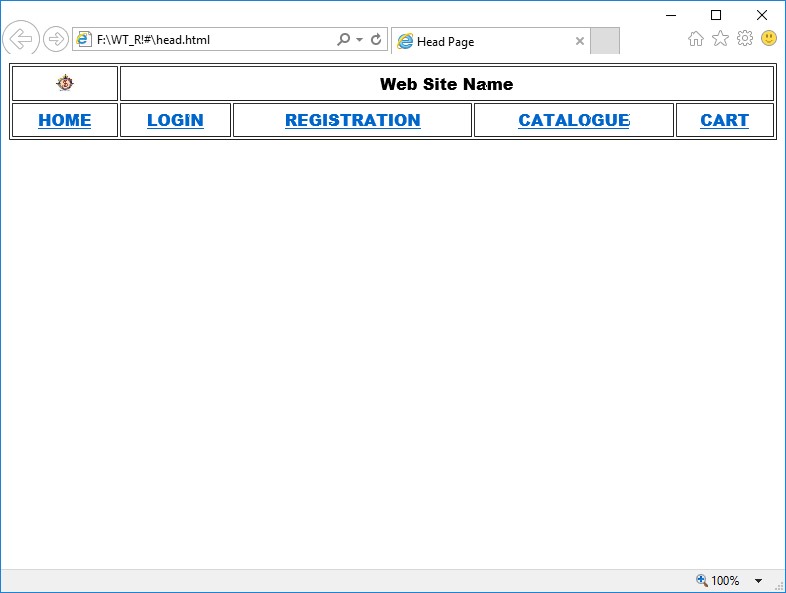
</table>

</font>

</body>

</html>

**OUTPUT:**



# dept.html

<html>

<head>

<title>Departments Page</title>

</head>

<body>

<font face="Arial Black" size="4">

<table align="center" height="100%">

<tr>

<td><a href="cat\_mca.html" target="des\_page">MCA</a></td>

</tr>

<tr>

<td><a href="cat\_mba.html" target="des\_page">MBA</a></td>

</tr>

<tr>

<td><a href="cat\_bca.html" target="des\_page">BCA</a></td>

</tr>

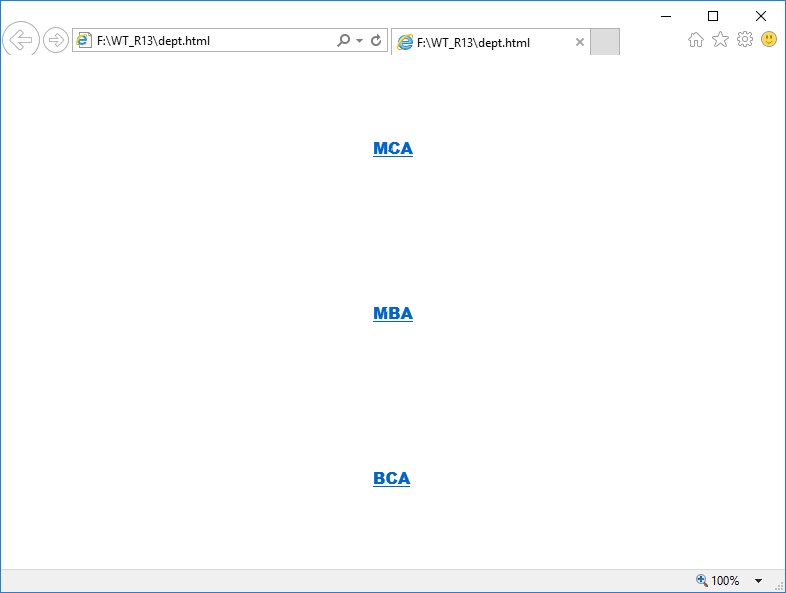
</table>

</font>

</body>

</html>

**OUTPUT:**



# desc.html

<html>

<head>

<title> Description page</title>

</head>

<body>

<br><br><br><br><br><br><br><br><br><br>

<font face="TIMES NEW ROMAN" size="5">

<center>

Description of the Website

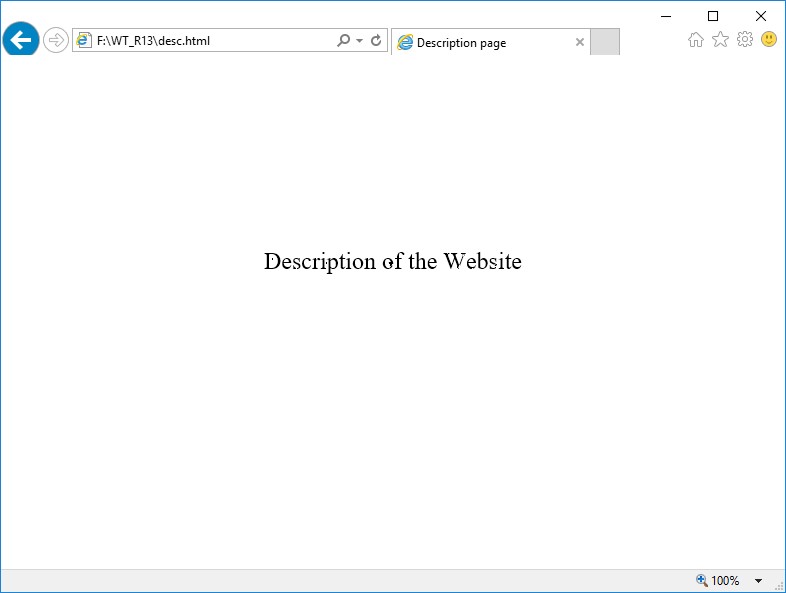
</center>

</font>

</body>

</html>

**OUTPUT:**



# home.html

<html>

<head>

<title>Home page</title>

</head>

<frameset rows="20,80">

<frame src="head.html" name="head\_page">

<frameset cols="15,85">

<frame src="dept.html" name="dept\_page">

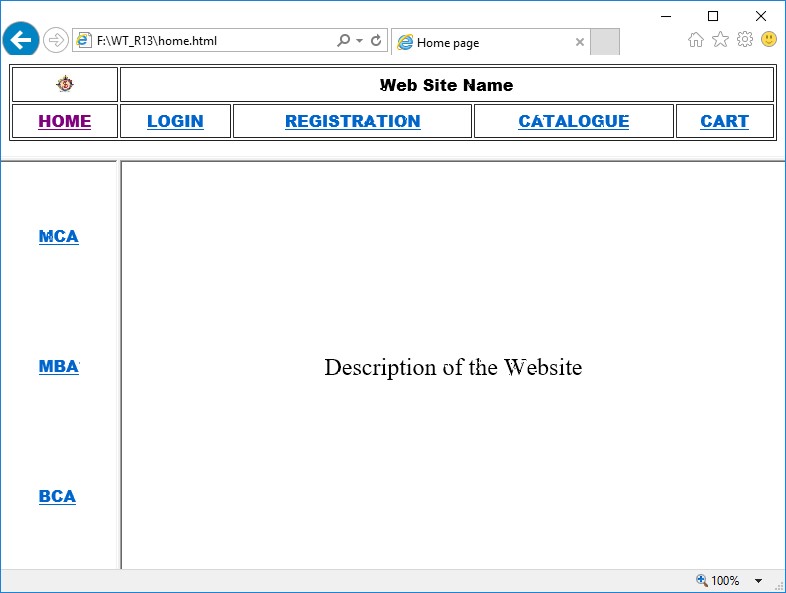
<frame src="desc.html" name="des\_page">

</frameset>

</frameset>

</html>

# OUTPUT



**2. LOGIN PAGE: Login page must contain Login field, Password field, Submit and reset buttons.**

# login.html

<html>

<head>

<title>Login page</title>

</head>

<body>

<center>

<font face="Arial Black" size="4"><u><b>LOGIN FORM</b></u></font>

<br><br>

<form name="f1" action="" method="post">

<table frame="box" cellspacing="10">

<tr>

<td>Login</td>

<td><input type="text" size="25"></td>

</tr>

<tr>

<td>Password</td>

<td><input type="password" size="25"></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value="SUBMIT">

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type="reset" value="RESET"></td>

</tr>

</table>

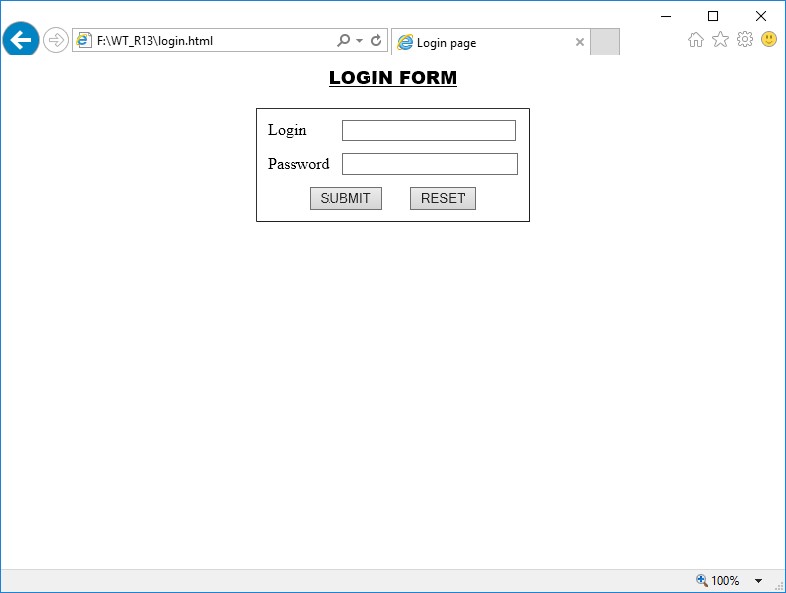
</form>

</center>

</body>

</html>

# OUTPUT



**3. CATALOGUE PAGE:**

The catalogue page should contain the details of all the books available in the web site in a table. The details should contain the following:

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

# catalogue.html

<html>

<head>

<title> Catalogue Page</title>

</head>

<body>

<table width="100%" height="100%" border="1">

<tr>

<td><img src="xmlb.gif" height="100" width="100"/></td>

<td>Book: XML Bible

<br>Author: Winston

<br>Publication: Wiely</td>

<td>$40.5</td>

<td><img src="adc.gif" height="100" width="100"/></td>

</tr>

<tr>

<td><img src="ai.gif" height="100" width="100"/></td>

<td>Book: AI

<br>Author: S.Russel

<br>Publication: Princeton hall</td>

<td>$63</td>

<td><img src="adc.gif" height="100" width="100"/></td>

</tr>

<tr>

<td><img src="java2.gif" height="100" width="100"/></td>

<td>Book: Java 2

<br>Author: Watston

<br>Publication: BPB publications</td>

<td>$35.5</td>

<td><img src="adc.gif" height="100" width="100"/></td>

</tr>

<tr>

<td><img src="html.gif" height="100" width="100"/></td>

<td>Book: HTML in 24 hours

<br>Author: Sam Peter

<br>Publication: Sam publications</td>

<td>$100</td>

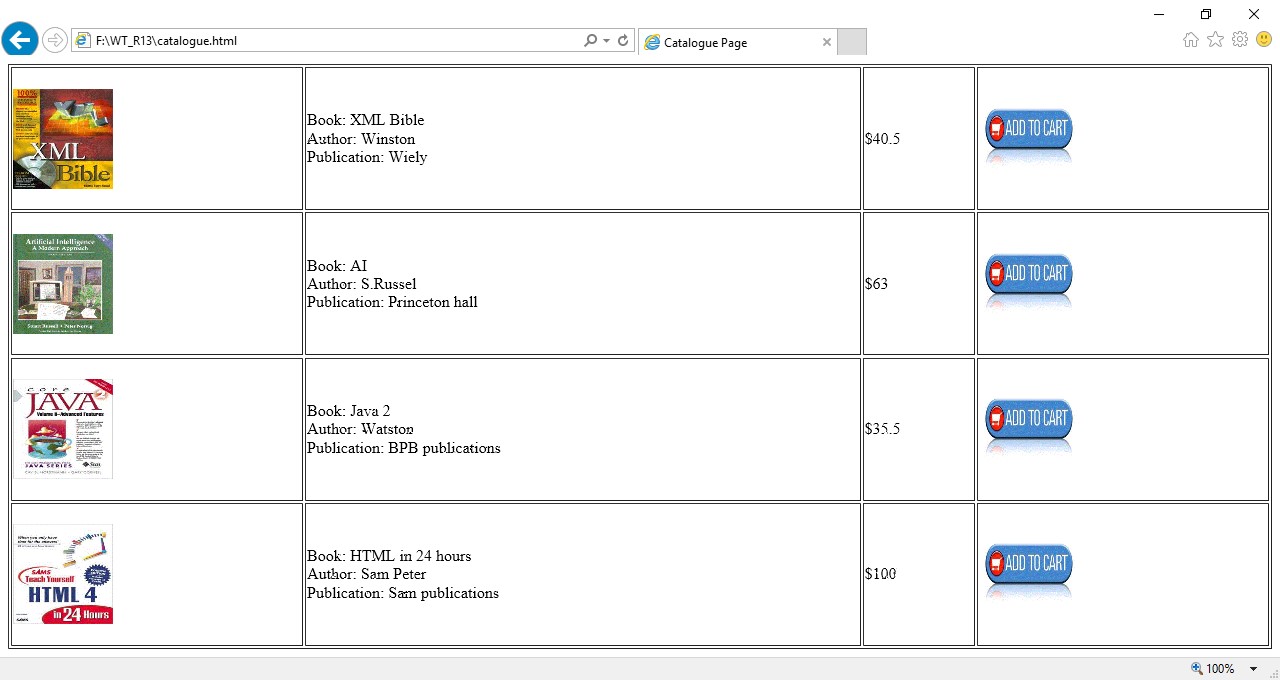
<td><img src="adc.gif" height="100" width="100"/></td>

</tr></table>

</body>

</html>

**OUTPUT:**



# cat\_mca.html

<html>

<head>

<title> MCA Catalogue Page</title>

</head>

<body>

<table width="100%" height="100%" border="1">

<tr>

<td><img src="xmlb.gif" height="100" width="100"/></td>

<td>Book: XML Bible

<br>Author: Winston

<br>Publication: Wiely</td>

<td>$40.5</td>

<td><img src="adc.gif" height="100" width="100"/></td>

</tr>

<tr>

<td><img src="html.gif" height="100" width="100"/></td>

<td>Book: HTML in 24 hours

<br>Author: Sam Peter

<br>Publication: Sam publications</td>

<td>$100</td>

<td><img src="adc.gif" height="100" width="100"/></td>

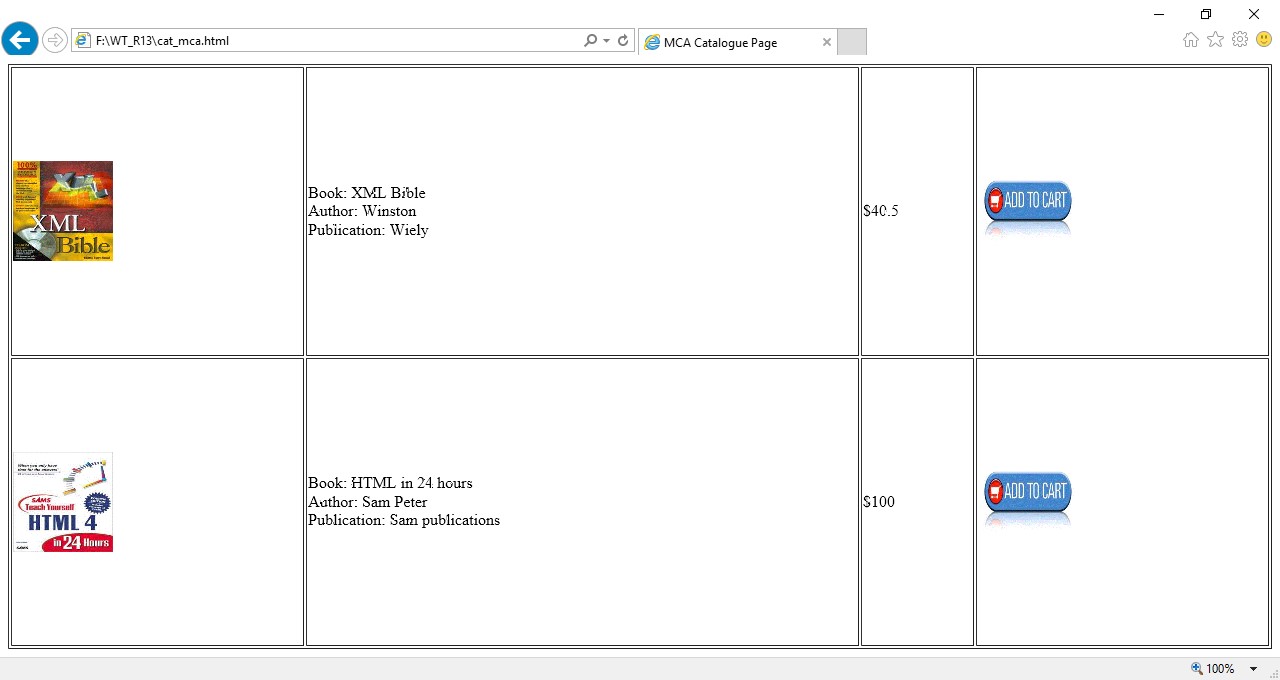
</tr>

</table>

</body>

</html>

**OUTPUT:**



# cat\_bca.html

<html>

<head>

<title> BCA Catalogue Page</title>

</head>

<body>

<table width="100%" height="100%" border="1">

<tr>

<td><img src="xmlb.gif" height="100" width="100"/></td>

<td>Book: XML Bible

<br>Author: Winston

<br>Publication: Wiely</td>

<td>$40.5</td>

<td><img src="adc.gif" height="100" width="100"/></td>

</tr>

<tr>

<td><img src="ai.gif" height="100" width="100"/></td>

<td>Book: AI

<br>Author: S.Russel

<br>Publication: Princeton hall</td>

<td>$63</td>

<td><img src="adc.gif" height="100" width="100"/></td>

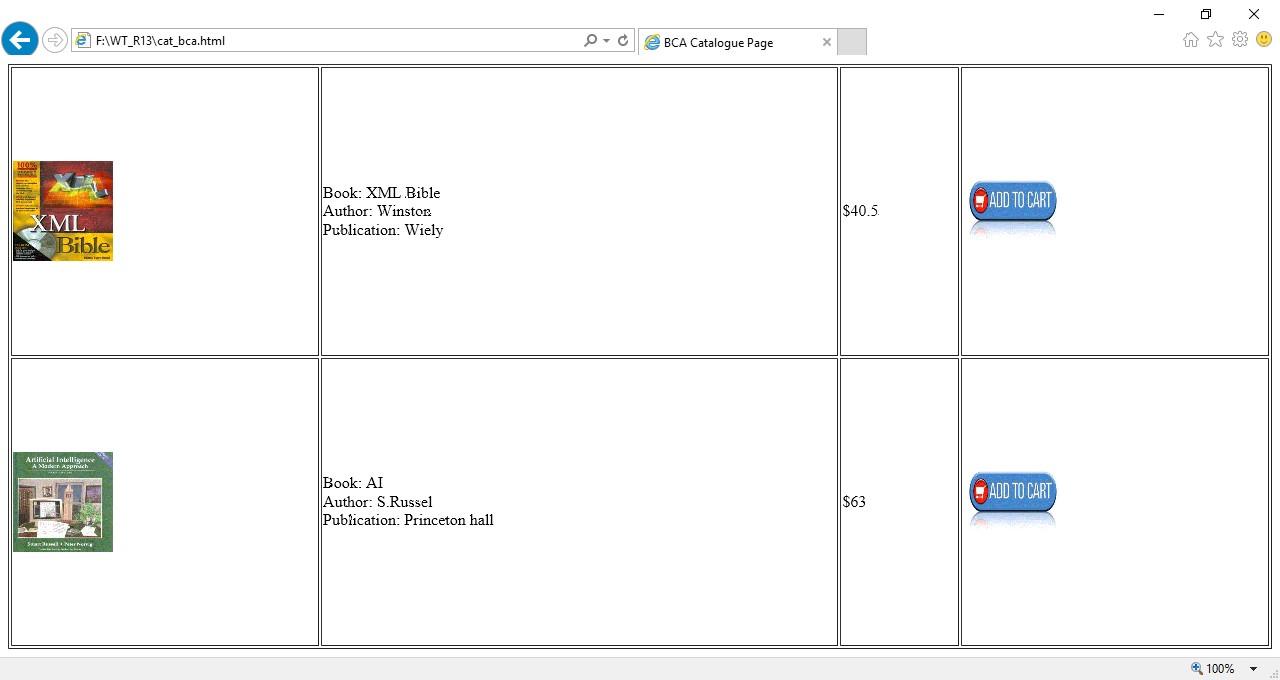
</tr>

</table>

</body>

</html>

**OUTPUT:**



**4. REGISTRATION PAGE:**

Create a “*registration form* “with the following fields

1. Name (Text field)
2. Password (password field)
3. E-mail id (text field)
4. Phone number (text field)
5. Sex (radio button)
6. Date of birth (3 select boxes)
7. Languages known (check boxes – English, Telugu, Hindi, Tamil)
8. Address (text area)

# registration.html

<html>

<head>

<title>Registration Page</title>

</head>

<body>

<center>

<h3 align="center"><u>REGISTRATION PAGE</u></h3>

<table border="3">

<tr><td>

<form name="f1" action="" method="post" onsubmit="">

<table cellspacing="10" cellpadding="5">

<tr><td>NAME</td><td><input type="text" size="30" name="uname"/></td></tr>

<tr><td>PASSWORD</td><td><input type="password" size="30" name="pass"/></td></tr>

<tr><td>E-MAIL ID</td><td><input type="text" size="30" name="email"/></td></tr>

<tr><td> PHONE NUMBER</td><td><input type="text" size="15" name="phone"/></td></tr>

<tr><td>GENDER</td><td><input type="radio" name="gen" value="m" />MALE

<input type="radio" name="gen" value="f" />FEMALE </td></tr>

<tr><td>DATE OF BIRTH</td>

<td><select name="day">

<option value="day">DAY</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

<option value="10">10</option>

<option value="11">11</option>

<option value="12">12</option>

<option value="13">13</option>

<option value="14">14</option>

<option value="15">15</option>

<option value="16">16</option>

<option value="17">17</option>

<option value="18">18</option>

<option value="19">19</option>

<option value="20">20</option>

<option value="21">21</option>

<option value="22">22</option>

<option value="23">23</option>

<option value="24">24</option>

<option value="25">25</option>

<option value="26">26</option>

<option value="27">27</option>

<option value="28">28</option>

<option value="29">29</option>

<option value="30">30</option>

<option value="31">31</option>

</select>

<select name="month">

<option value="month">MONTH</option>

<option value="jan">JANUARY</option>

<option value="feb">FEBRUARY</option> <option value="mar">MARCH</option> <option value="apr">APRIL</option>

<option value="may">MAY</option>

<option value="jun">JUNE</option>

<option value="jul">JULY</option>

<option value="aug">AUGUST</option>

<option value="sep">SEPTEMBER</option>

<option value="oct">OCTOBER</option>

<option value="nov">NOVEMBER</option> <option value="dec">DECEMBER</option>

</select>

<select name="year">

<option value="year">YEAR</option>

<option value="1986">1986</option>

<option value="1987">1987</option>

<option value="1988">1988</option>

<option value="1989">1989</option>

<option value="1990">1990</option>

<option value="1991">1991</option>

<option value="1992">1992</option>

<option value="1993">1993</option>

<option value="1994">1994</option>

<option value="1995">1995</option>

<option value="1996">1996</option>

<option value="1997">1997</option>

<option value="1998">1998</option>

<option value="1999">1999</option>

<option value="2000">2000</option>

<option value="2001">2001</option>

<option value="2002">2002</option>

<option value="2003">2003</option>

<option value="2004">2004</option>

<option value="2005">2005</option>

<option value="2006">2006</option>

<option value="2007">2007</option>

<option value="2008">2008</option>

<option value="2009">2009</option>

<option value="2010">2010</option>

<option value="2011">1980</option>

<option value="2012">1981</option>

<option value="2013">1982</option>

<option value="2014">1983</option>

</select></td></tr>

<tr><td>LANGUAGES KNOWN</td>

<td>

<input type="checkbox" value="eng" name="lang" />ENGLISH

<input type="checkbox" value="tel" name="lang" />TELUGU

<input type="checkbox" value="hin" name="lang" />HINDI

<input type="checkbox" value="tam" name="lang" />TAMIL

</td></tr>

<tr>

<td>ADDRESS</td>

<td><textarea name="addr" cols="25" rows="5"></textarea></td></tr>

<tr><td colspan="2" align="center"><input type="submit" value="SUBMIT"/>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="RESET" /></td>

</tr>

</table> </form>

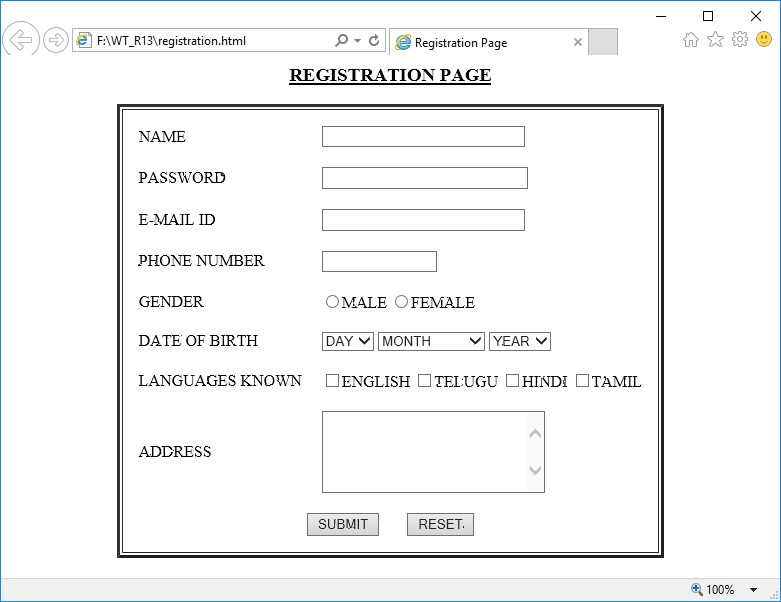
</td></tr></table>

</center>

</body>

</html>

**OUTPUT:**



**5. Design a web page using CSS (Cascading Style Sheets) which includes the following:**

**A. Use different font, styles: In the style definition you define how each selector should work (font, color etc.). Then, in the body of your pages, you refer to these selectors to activate the styles.**

# styles.css

|  |  |
| --- | --- |
| h1  { |  |
|  | color:red; |
|  | font-family:caStellar; |
|  | font-size:22pt; |
| }  h2  { | text-decoration:underline; |
|  | color:blue; |
|  | font-family:Chiller; |
|  | font-size:18pt; |
| }  p  { | text-decoration:overline; |
|  | color:magenta; |
|  | font-family:Trebuchet MS; |
|  | font-size:14pt; |
| } | font-style:italic; |

# fontstyles.html

<html>

<head>

<title>Usage of different font,styles and colors </title>

<link rel="stylesheet" type="text/css" href="styles.css"/>

</head>

<body>

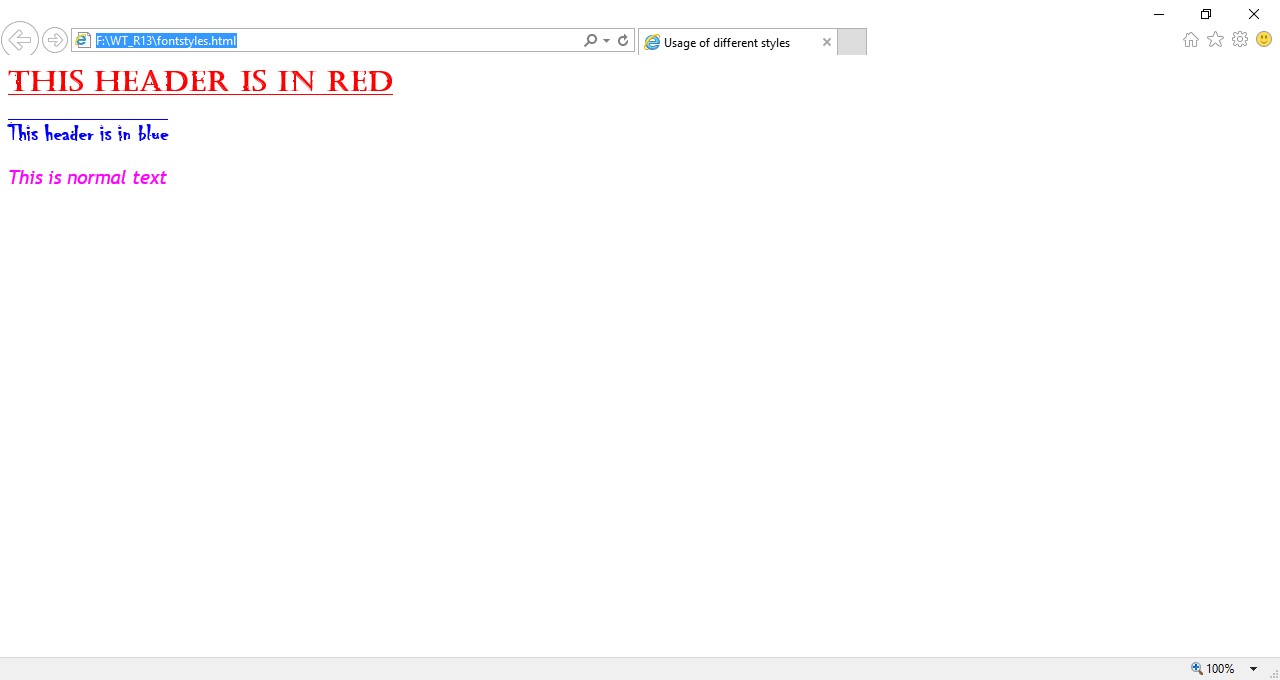
<h1>This header is in red</h1>

<h2>This header is in blue</h2>

<p>This is normal text</p>

</body></html>

**OUTPUT:**



**B. Set a background image for both the page and single elements on the page.**

# bg\_image.html

<html>

<head>

<title>Setting background image</title>

<style type="text/css">

body {

background-image:url("img11.jpg");

}

</style>

</head>

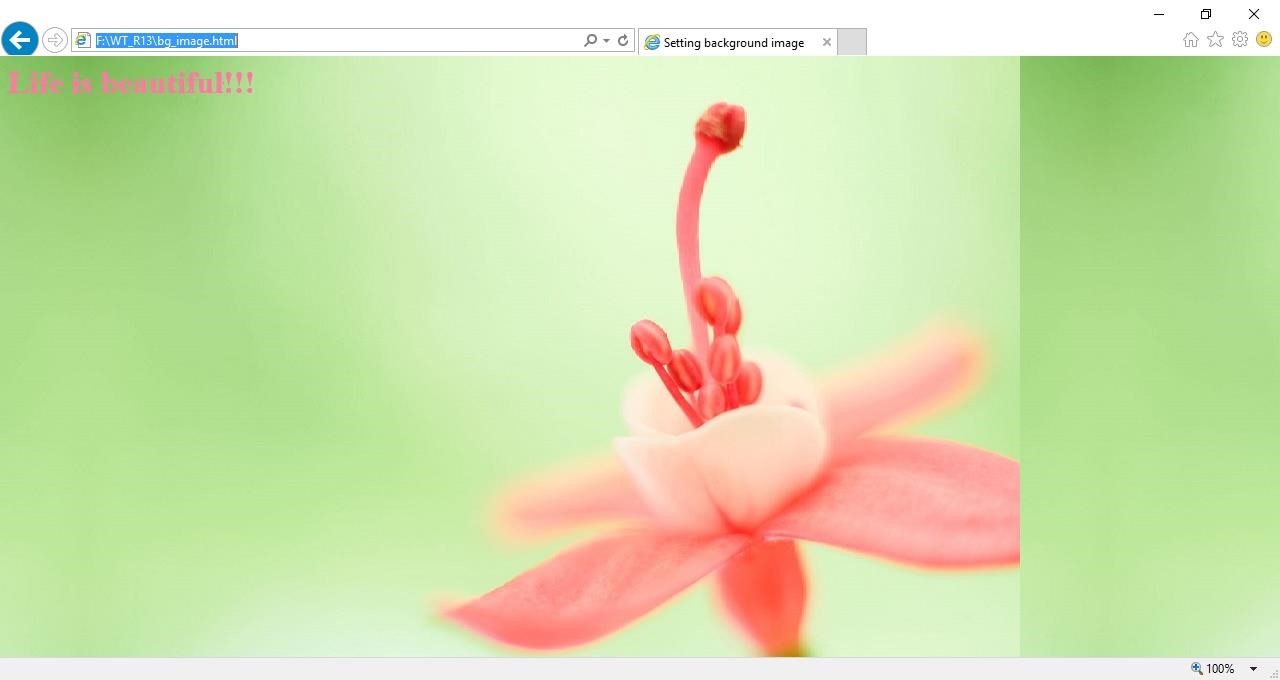
<body text="#ee78a2">

<h1>Life is beautiful!!!</h1>

</body>

</html>

**OUTPUT:**



**C. Control the repetition of the image with the background-repeat property.**

# bg\_repeat.html

<html>

<head>

<title>Controlling background image</title>

<style type="text/css">

body {

background-image:url("img11.jpg");

background-repeat:no-repeat

}

</style>

</head>

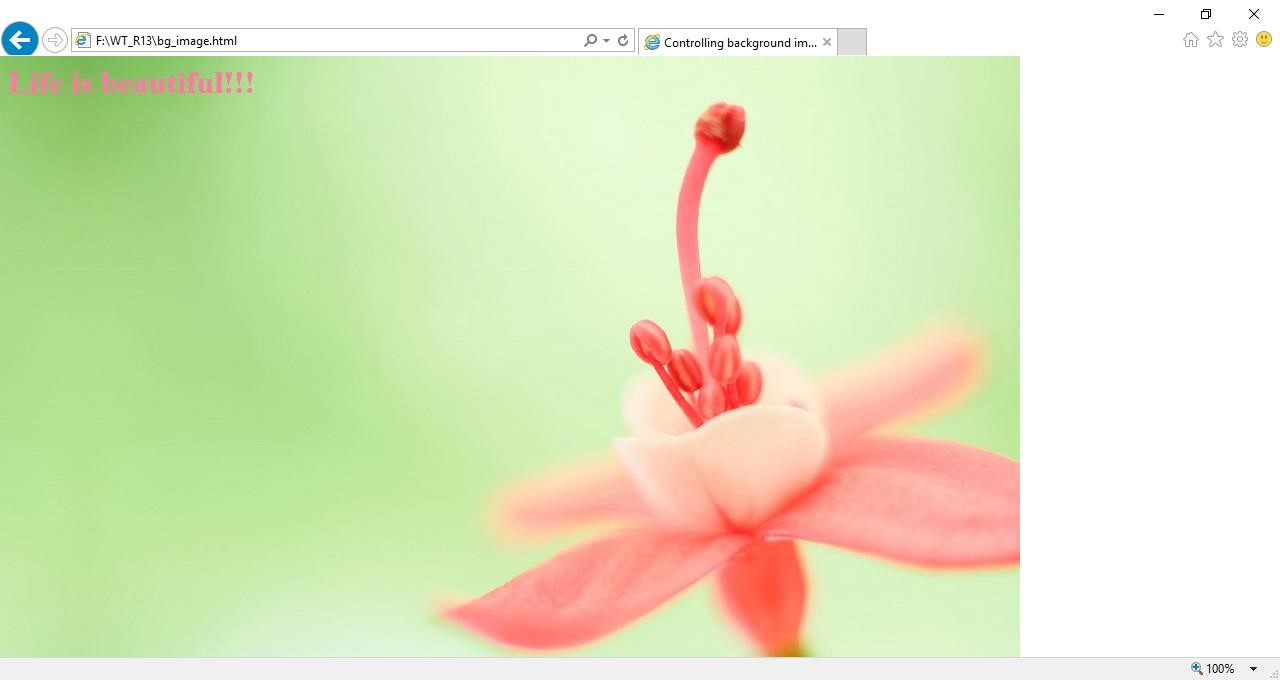
<body text="#fe78a2">

<h1>Life is beautiful!!!</h1>

</body>

</html>

# OUTPUT



**6.** Write an XML file which will display the Book information which includes the following:

1. Title of the book
2. Author Name
3. ISBN number
4. Publisher name
5. Edition
6. Price

Write a Document Type Definition (DTD) to validate the above XML file.

**catalog.dtd**

<!ELEMENT Catalog (Book)\*>

<!ELEMENT Book (Title, Author, Publication ,Edition, ISBN, Price)>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Publication (#PCDATA)>

<!ELEMENT Edition (#PCDATA)>

<!ELEMENT ISBN (#PCDATA)>

<!ELEMENT Price (#PCDATA)>

# library.css

Catalog

{

|  |  |
| --- | --- |
|  | font-family:arial; |
|  | color:blue; |
|  |  |
| }  Book  { | font-size:16pt |
|  |  |
|  | font-family:times new roman; |
|  | color:blue; |
|  | table-layout:auto; |
| }  Title  { | font-size:14pt |
|  | font-family:arial; |
|  | color:green; |
|  | margin-left:20pt; |
|  | font-size:12pt |

}

Author

{

font-family:arial;

color:magenta

}

Publication,Edition,ISBN,Price

{

display:block;

font-family:arial; color:black; font-size:10pt;

margin-left:40pt

}

# Cataloguedemo.xml

<?xml version="1.0" encoding="UTF-8" standalone=”no”?>

<?xml-stylesheet type="text/css" href="library.css" ?>

<!DOCTYPE Catalog SYSTEM "catalog.dtd">

<Catalog">

<Book>

<Title>XML Bible</Title>

<Author>Winston</Author>

<Publication>Wiely</Publication>

<Edition>Fifth Edition</Edition>

<ISBN>0-7645-4760-7</ISBN>

<Price>$40.5</Price >

</Book>

<Book>

<Title>Artificial Intelligence</Title>

<Author>S. Russel</Author>

<Publication> Princeton Hall </Publication>

<Edition> Sixth Edition</Edition>

<ISBN> 0-13-1038-5-2 </ISBN>

<Price>$63</Price>

</Book>

<Book>

<Title>Java 2</Title>

<Author>Watson</Author>

<Publication>BPB Publications</Publication>

<Edition>Third Edition</Edition>

<ISBN>0-41-1058-7-2</ISBN>

<Price>$63</Price>

</Book>

<Book>

<Title>HTML in 24 hours </Title>

<Author> Sam Peter</Author>

<Publication> SAM Publications </Publication>

<Edition>Fifth Edition</Edition>

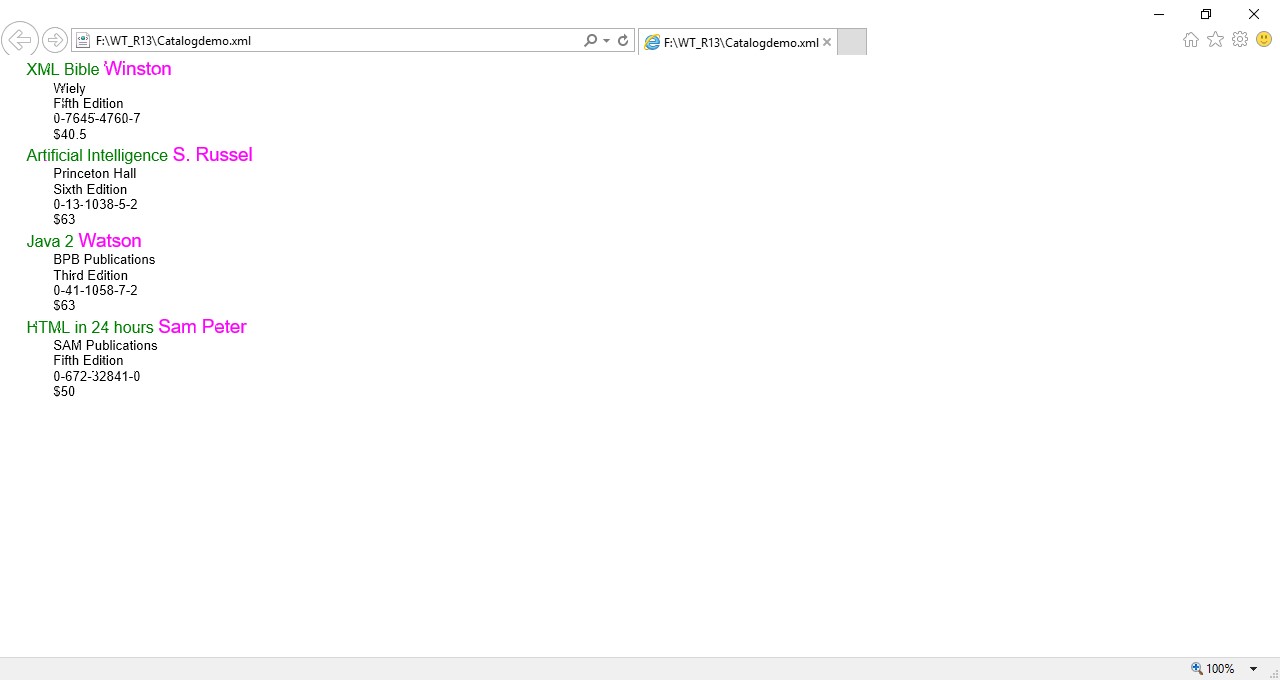
<ISBN> 0-672-32841-0 </ISBN>

<Price> $50 </Price >

</Book>

</Catalog>

**OUTPUT:**



**7. Write Ruby program reads a number and calculates the factorial value of it and prints the same.**

# fact.rb

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



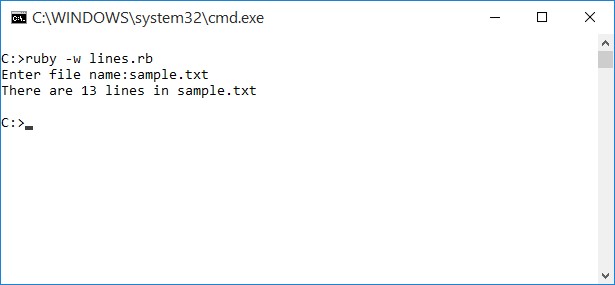
**8. Write a Ruby program which counts number of lines in a text file using its regular expressions facility**

# lines.rb

print "Enter file name:"; fname = gets.chomp; count = IO.readlines(fname).size

puts "There are #{count} lines in #{fname}";

**OUTPUT:**



**9. Write a Ruby program that uses iterator to find out the length of a string.**

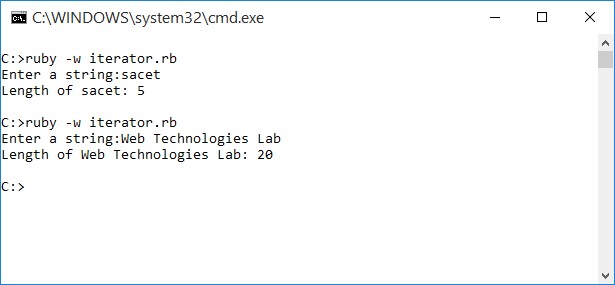
# iterator.rb

print "Enter a string:" str = gets.chomp c=0

str.each\_char do |i| c=c+1 end

puts "Length of #{str}: #{c}"

# OUTPUT



**10. Write simple Ruby programs that uses arrays in Ruby.**

# bsort.rb

puts "Bubble Sort"

puts "==========="

print "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

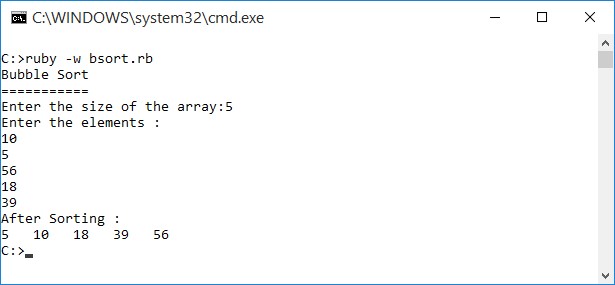
end end

end

puts "After Sorting :" for i in 0..n-1

print "#{a[i]} " end

**OUTPUT:**



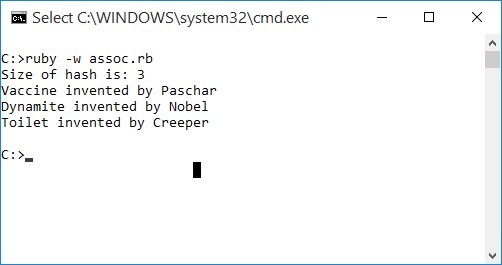
**11. Write programs which uses associative arrays concept of Ruby.**

# assoc.rb

creators = Hash.new;

creators = {"Vaccine"=>"Paschar","Dynamite"=>"Nobel","Toilet"=>"Creeper"}; size = creators.length; puts "Size of hash is: #{size}"; creators.each do |key,val| puts "#{key} invented by #{val}" end

**OUTPUT:**



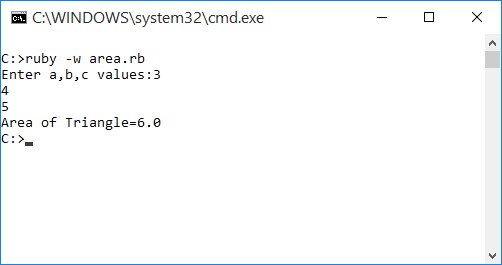
**12. Write Ruby program which uses Math module to find area of a triangle.**

# area.rb

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:**



**13. Write Ruby program which uses tk module to display a window**

# window.rb

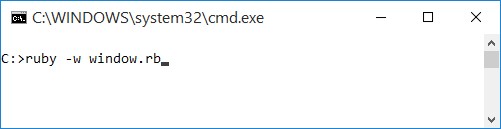
require 'tk'

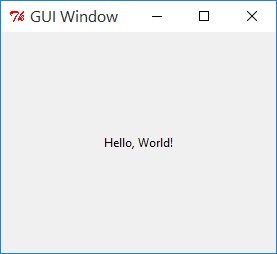
root = TkRoot.new { title "GUI Window" }

TkLabel.new(root) do text 'Hello, World!'

pack("side" => "right", "padx"=> "100", "pady"=> "100") end Tk.mainloop

**OUTPUT:**





**14. Define Complex class in Ruby and do write methods to carry operations on Complex objects.**

# Cmplx.rb

class Cmplx

attr\_accessor :real, :imag

def read

@real=gets.to\_i @imag=gets.to\_i end

def add(other)

ob = Cmplx.new

ob.real = @real+other.real;

ob.imag = @imag+other.imag;

return ob

end

def subtract(other)

ob = Cmplx.new;

ob.real = @real-other.real;

ob.imag = @imag-other.imag; return ob; end

def multiply(other)

ob = Cmplx.new

ob.real = (@real \* other.real)-(imag\*other.imag);

ob.imag = (@real \* other.imag)+(imag\*other.real);

return ob; end

def divide(other)

t = Cmplx.new; ob = Cmplx.new;

t.imag = -other.imag; r =(other.real).abs; i =(other.imag).abs;

d =(r\*r)+(i\*i);

ob.real = ((@real \* other.real)-(imag\*t.imag))/d; ob.imag = ((@real \* t.imag)+(imag\*other.real))/d;

return ob; end

def disp

if @imag<0

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | puts "#{@real}#{@imag}i" |
|  |  | else |  |
|  |  |  | puts "#{@real}+#{@imag}i" |
|  |  | end |  |
| end | end |  |  |

t1 = Cmplx.new t2 = Cmplx.new t3 = Cmplx.new t4 = Cmplx.new t5 = Cmplx.new t6 = Cmplx.new

puts "Enter first number" t1.read

puts "Enter Second number"

t2.read

print "FIRST NUMBER:" t1.disp

print "SECOND NUMBER:"

t2.disp

print "=================\n";

print "ADDITION:" t3 = t1.add(t2)

t3.disp

print "SUBTRACTION:"

t4 = t1.subtract(t2) t4.disp

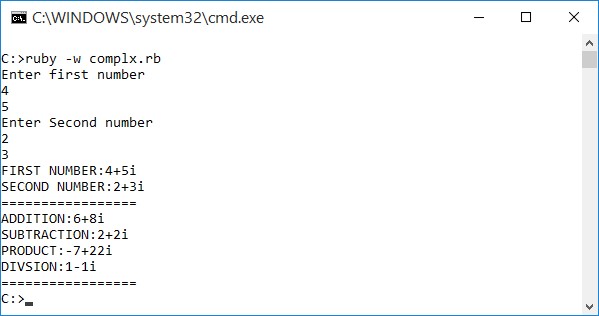
print "PRODUCT:"

t5 = t1.multiply(t2) t5.disp

print "DIVSION:" t6 = t1.divide(t2) t6.disp

print "=================";

**OUTPUT:**



**15. Write a program which illustrates the use of associative arrays in Perl.**

# pAssoc.pl

%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 adding element:\n"; print"===========================\n";

while(($key) = each %ages)

{

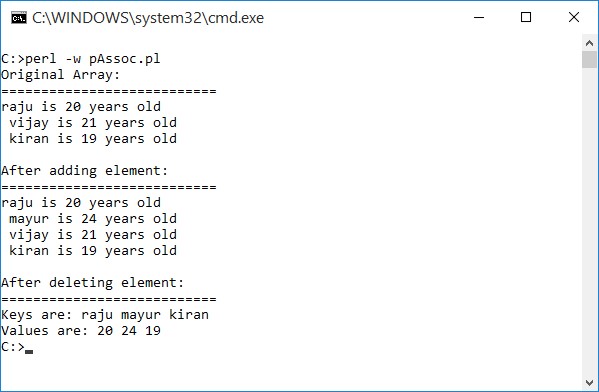
print "$key is $ages{$key} years old\n ";

}

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:**



**16. Write Perl program takes a set names along the command line and prints whether they are regular files or special files**

# fileTest.pl

$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

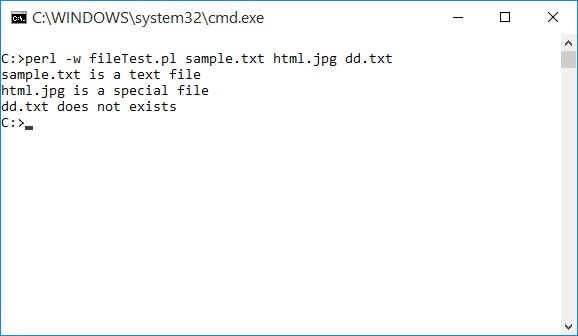
{

print "$ARGV[$i] does not exists";

}

}

**OUTPUT:**



**17. Write a Perl program to implement UNIX „*passwd’* program**

# unixPw.pl

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 )

{

$salt = join('',@saltchars[rand(64),rand(64)]);

}

else

{

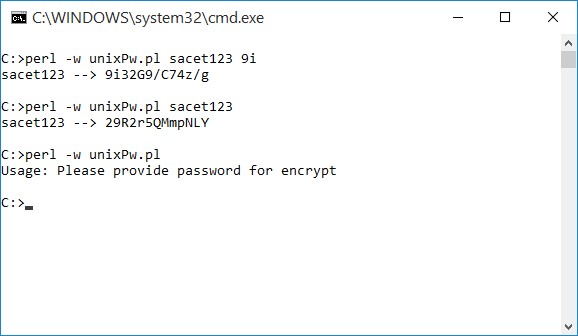
$salt=$ARGV[1];

}

$encrypted=crypt($password,$salt);

print "$password --> $encrypted\n";

**OUTPUT:**



**18. An example Perl program to connect to a MySQL Database table and executing simple commands.**

**Note:**  Create the following table in MySQL and insert some data in to the table.

students table

create table students

(

roll varchar(3) not null, name varchar(3) not null,

branch varchar(3) not null

);

# dbConn.pl

use DBI;

#definition of variables

$db="test";

$host="localhost";

$user="root";

$password="";

#connect to MySQL database

my $dbh = DBI->connect ("DBI:mysql:database=$db:host=$host",$user,$password) or die "Can't connect to database: $DBI::errstr\n";

#prepare the query

my $sth = $dbh->prepare( "SELECT \*FROM students");

#execute the query

$sth->execute();

## Retrieve the results of a row of data and print print "\tQuery results:\n=====================\n"; print("ROLL\tNAME\tBRANCH\n=====================\n"); while (my @row = $sth->fetchrow\_array())

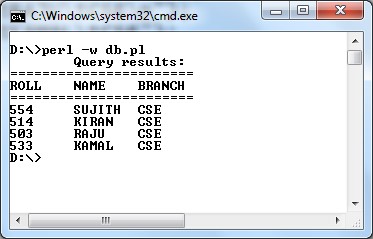
{

print "@row\n";

}

warn "Problem in retrieving results", $sth->errstr( ), "\n" if $sth->err(); exit;

**OUTPUT:**



**19. Example PHP program for contacts page.**

# contacts.php

<?php

if(isset($\_REQUEST['vname']))

{

$vname=$\_REQUEST['vname'];

$email=$\_REQUEST['email'];

$message=$\_REQUEST['message'];

$from="From: $vname<$email>\r\nReturn-path: $email";

$subject="Message sent using your contact form";

$m = mail("youremail@yoursite.com", $subject, $message, $from); if($m)

echo "<h1>Email sent!";

else

echo "Send Failed:".mysql\_error();

echo "&nbsp;&nbsp;&nbsp;<a href='contacts.php'>[Back to Contacts]</a>";

} else { ?>

<html>

<head>

<title>Contacts Page</title>

</head>

<body>

<font face="Trebuchet ms" size="6">

<table border="1">

<tr><th>Contact Us</th></tr>

<tr><td>

<form action="contacts.php" method="POST" enctype="multipart/form-data"> Your name<br/>

<input type="text" name="vname" value="" size="30"/><br>

Your email<br>

<input type="text" name="email" value="" size="30"/><br>

Your message<br>

<textarea name="message" rows="7" cols="30"></textarea><br>

<input type="submit" value="Send email"/>

</form>

</td></tr>

</table>

</body>

</html>

<?php } ?>

**OUTPUT:**

