ASSIGNMENT

1 ) Find occurrence of RUBY from string1.

**irb(main):011:0> bhabani1.scan(/RUBY/)**

**=> ["RUBY", "RUBY", "RUBY"]**

**irb(main):012:0> bhabani1.scan(/RUBY/).count()**

**=> 3**

2) Find the position where RUBY occurs in String1

**=>Bhabani\_num\_for(:scan,/RUBY/).map{Regexp.last\_match.begin(0)}**

**=> [0, 114, 195]**

3) Create an array of words using string1 and print them using recursive function

**irb(main):016:0> arr5=bhabani\_string1.split(' ')**

**["RUBY", "parses", "a", "file", "by", "looking", "for", "<br/>", "one", "of", "the", "special", "tags", "that", "tells", "it", "to", "start", "interpreting", "the", "text", "as", "RUBY", "code.", "The", "parser", "then", "executes", "all", "of", "the", "code", "it", "finds", "until", "it", "runs", "into", "a", "RUBY", "closing", "<br/>", "tag."]**

**irb(main):023:0> arr.each do |i|**

**irb(main):024:1\* puts i**

**irb(main):025:1> end**

**RUBY**

**parses**

**a**

**file**

**by**

**looking**

**for**

**<br/>**

**one**

**of**

**the**

**special**

**tags**

**that**

**tells**

**it**

**to**

**start**

**interpreting**

**the**

**text**

**as**

**RUBY**

**code.**

**The**

**parser**

**then**

**executes**

**all**

**of**

**the**

**code**

**it**

**finds**

**until**

**it**

**runs**

**into**

**a**

**RUBY**

**closing**

**<br/>**

**tag.**

**=> ["RUBY", "parses", "a", "file", "by", "looking", "for", "<br/>", "one", "of", "the", "special", "tags", "that", "tells", "it", "to", "start", "interpreting", "the", "text", "as", "RUBY", "code.", "The", "parser", "then", "executes", "all", "of", "the", "code", "it", "finds", "until", "it", "runs", "into", "a", "RUBY", "closing", "<br/>", "tag."]**

4) Capitalise string1

Solution:

**=>bhabani1=”RUBY parses a file by looking for </br/> one of the special tags that tells it to start**

**interpreting the text as RUBY code.The parser then executes all of the code it finds until**

**it runs into a RUBY closing <br/>tag.”**

**=>bhabani1.capitalize**

**Model example:**

**=>abc=”hello”**

**=>abc.capitalize**

**=>Hello**

5) Combine String1 and String2

**=>Bhabani1=”string1”**

**=>Bhabani3=”string3”**

**=>Bhabani1+bhabani3**

**Alternative Method:**

**Bhabani1.concat(bhabani3)**

**Method**:

str + other\_str

**str.concat(other\_str)**

Concatenates other\_str to str.

Solution:

=>bhabani1=”

**Solution:1**

**=>bhabani1=”Hello”**

**=>bhabani2=”GoodMorning”**

**=>name1+name2**

**=>HelloGoodMorning**

**Sloution2:**

**=> bhabani1=”Hello”**

**=>bhabani2=”GoodMorning”**

**=>bhabani1.concat(bhabani2)**

**=>HelloGoodMorning**

6) Print string1 and string 2 using here doc

**irb(main):010:0> string1=<<-"here document ends"**

**< finds until it runs into a RUBY closing <br/> tag.**

**irb(main):003:0" here document ends**

**=> "RUBY parses a file by looking for <br/> one of the special tags that tells i**

**t to start interpreting the text as RUBY code. The parser then executes all of t**

**he code it finds until it runs into a RUBY closing <br/> tag.\n"**

**irb(main):004:0> string2=<<-"here document ends"**

**<ariable's type is determined by the context in which the variable is used.**

**irb(main):006:0" here document ends**

**=> "RUBY does not require (or support) explicit type definition in variable decl**

**aration; a variable's type is determined by the context in which the variable is**

**used.\n"**

**irb(main):011:0>**

7) Print current date

**=>bhabani=Time.now()**

**=>bhabani.strftime(“The date is %d-%m-%y”)**

**=> bhabani.strftime(“The date is %m-%d-%y”)**

**=>bhabani.strftime(“The date is %d-%B-%y”)**

**=>bhabani.strftime(“The date is %a-%B-%y”)**

**=>bhabani.strftime(“The date is %a-%B-%y”)**

8) Print 12th January 2012

**=>bhabani=Time.now()**

**=>bhabani=strftime(“12 -Jan - %Y”)**

9) Add 7 days to the current date

Solution:

**>> date = Date.today**

**>> date.to\_s**

**=> "2012-12-13"**

**>> Next\_set\_date = date + 7**

**>> Next\_set\_date.to\_s**

**=> "2012-12-20"**

10) Cut the string 1 into 4 parts and print it

**Solution:**

**Bhabani1 =”RUBY parses a file by looking for </br/> one of the special tags that tells it to start**

**interpreting the text as RUBY code.The parser then executes all of the code it finds until**

**it runs into a RUBY closing <br/>tag.”**

**=>bhabani1.split(‘ ‘,4)**

12 ) Remove the HTML characters from string

14) Find the length of string1 and string2

**Solution:bhabani1:length**

**ing for </br/> one of the special tags that tells it to start**

**he parser then executes all of the code it finds until**

**rb(main):003:0" it runs into a RUBY closing <br/>tag."**

**> "RUBY parses a file by looking for </br/> one of the special tags that tells**

**it to start\ninterpreting the text as RUBY code.The parser then executes all of**

**the code it finds until\nit runs into a RUBY closing <br/>tag."**

**rb(main):005:0> bhabani1.class**

**> String**

**rb(main):006:0> bhabani1.length**

**> 217**

**String length 2:**

**=>Bhabani2=”whole content to be written”**

**=>bhabani2.class**

**=>bhabani2.length**

**=>162**

16) Print all Global variables provided by ruby

17) Usage and example of Header Ruby

The standard Ruby library provides a tool for generating documentation from self documenting code called RubyDoc or RDoc for short. Though it's part of the standard library, RDoc isn't a file you require to use. Instead it consists of a command-line program called rdoc and a library for parsing specially-formatted comments out of Ruby and C code.

Example :

18) Redirect page 1 to page 2**(doubt**)

19) **Compare two dates (12-4-2010 &12-5-2011)**

Solution:

**=>require 'date'**

**=>past=Date.new(2010,4,12)**

**=>present=Date.new(2011,5,12)**

**=>time\_until=present-past**

**=>time.until.to\_i**

**=>395 days**

**=>**

20) Print date after 20 days after current date

Solution:

**>> date = Date.today**

**>> date.to\_s**

**=> "2012-12-13"**

**>> Next\_set\_date = date + 20**

**>> Next\_set\_date.to\_s**

**=> "2013-01-02"**

21) Print date in array format