Quiz 4

1. Ruby Program that prints the Time in different time zones
Ans:

```
puts "enter time zone code 1 for california or 2 for BD"
time_zone = gets.to_i
case time_zone
 when 1
 puts Time.now
 when 2
 puts now.gmtime
 end
```

2. Write a program that iterates over an array and builds a new array that is the result of incrementing each value in the original array by a value of 2. You should have two arrays at the end of this program,

The original array and the new array you've created.Print both arrays to the screen using the p method instead of puts.

Ans:

```
array = [3, 1, 5, 4, 2]

new_array = []

array.each do |a|

 new_array<<a + 2

end

p array

p new_array
```

3. Ruby program to find the leap year when start and end year are given.

Ans:

```
puts 'Input a starting year:'
```

```
start_year = gets.chomp
puts 'Input an ending year:'
end_year = gets.chomp
puts 'The leap year are'

while start_year.to_i <= end_year.to_i
if start_year.to_f%400 == 0
puts start_year
elsif start_year.to_f%100 == 0
elsif start_year.to_f%4 == 0
puts start_year
end
start_year = start_year.to_i + 1
end
```

4. Ruby program that takes a numerical value and give the output as Roman number

5. Write a your own ruby program that uses a Queue

Ans:

```
require 'thread'

queue = Queue.new

producer = Thread.new do
  5.times do |i|
    sleep rand(i) # simulate expense
```

```
    queue << i
    puts "#{i} produced"
  end
end

consumer = Thread.new do
  5.times do |i|
    value = queue.pop
    sleep rand(i/2) # simulate expense
    puts "consumed #{value}"
  end
end

consumer.join
```

6. Write your own ruby program that uses each_with_index method to iterate through an array that prints each index and value

Ans:

```
top_five_games = ["mario brothers", "excite bike","ring king","castlevania","double dragon"]

top_five_games.each_with_index do | game, index |
 puts "#{index + 1}. #{game}"
end
```

7. Ruby Program that prints if duplicates existing in a array

Ans:

```
def find_duplicates(elements)
   encountered = {}
```

```
  # Examine all elements in the array.
  elements.each do |e|
   # If the element is in the hash, it is a duplicate.
   if encountered[e]
     puts "Dupe exists for: " << e
   else
     # Record that the element was encountered.
     encountered[e] = 1
   end
  end
end
```

8. Write a Ruby program that prints pascal triangle

Ans:

```
def find_num(n, k)
  result = factorial(n) / (factorial(k) * factorial(n - k))
end

def pascale(num)
i = 0
scale = 75
  while i <= num
    new_arr = []
   (0..i).map {|x| new_arr << find_num(i, x)}
    p new_arr.to_s.rjust(50 + scale)
```

```
   i += 1
   scale += 1
 end

def factorial(num)
 if num == 0
  return 1
 else
  num *= factorial(num - 1)
 end
end


end

pascale(12)
```

9. Write a Ruby program that prints the length of the common string when two strings are compared.

```
def substing(s1, s2)

      return 0 if s1.empty? || s2.empty?

      num=Array.new(s1.size){Array.new(s2.size)}
      s1.scan(/./).each_with_index{|letter1,i|
         s2.scan(/./).each_with_index{|letter2,j|

               if s1[i]==s2[j]
                  if i==0||j==0
                    num[i][j] = 1
                  else
                    num[i][j]  = 1 + num[i - 1][ j - 1]
                  end
```

```
            else
               if i==0 && j==0
                  num[i][j] = 0
               elsif i==0 &&  j!=0  #First ith element
                  num[i][j] = [0,  num[i][j - 1]].max
               elsif j==0 && i!=0  #First jth element
                   num[i][j] = [0, num[i - 1][j]].max

               elsif i != 0 && j!= 0
                 num[i][j] = [num[i - 1][j], num[i][j - 1]].max
               end
            end
         }
      }
      num[s1.length - 1][s2.length - 1]

end

puts substing("room","roommate")
```