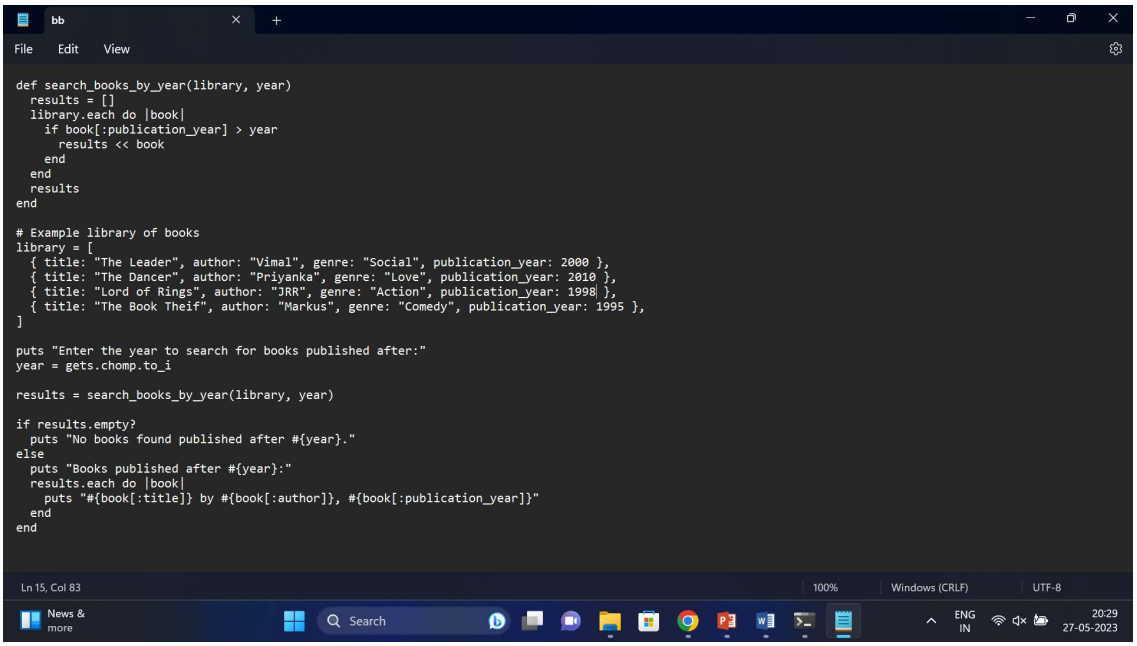
RUBY LAB ASSIGNMENT 3.1

21MIS1021

VIMAL KUMAR S

1. Scenario: Managing a Library Catalog Question: You are building a library catalog system in Ruby. Each book has multiple attributes such as title, author, genre, and publication year. Design a hash structure to store information about multiple books and implement a method to search for books published after a specific year.
   CODE:




```
def search_books_by_year(library, year)
 results = []
 library.each do |book|
  if book[:publication_year] > year
   results << book
  end
 end
 results
end

# Example library of books
library = [
 { title: "The Leader", author: "Vimal", genre: "Social", publication_year: 2000 },
 { title: "The Dancer", author: "Priyanka", genre: "Love", publication_year: 2010 },
 { title: "Lord of Rings", author: "JRR", genre: "Action", publication_year: 1998 },
 { title: "The Book Theif", author: "Markus", genre: "Comedy", publication_year: 1995 },
]
```

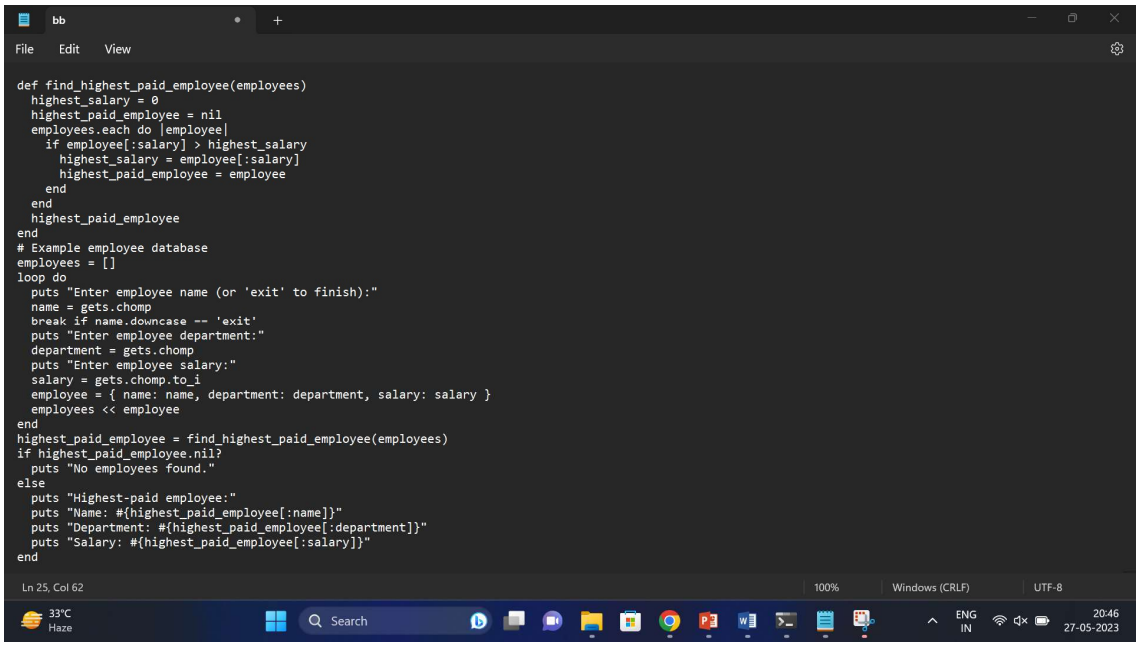
```
puts "Enter the year to search for books published after:"
year = gets.chomp.to_i

results = search_books_by_year(library, year)

if results.empty?
 puts "No books found published after #{year}."
else
 puts "Books published after #{year}:"
 results.each do |book|
  puts "#{book[:title]} by #{book[:author]}, #{book[:publication_year]}"
 end
end
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021 VIMAL KUMAR S>ruby bb.rb
Enter the year to search for books published after:
1999
Books published after 1999:
The Leader by Vimal, 2000
The Dancer by Priyanka, 2010
```

2. Scenario: Employee Database Question: You are developing an employee management system. Design a hash structure to store information about employees, including their names, departments, and salaries. Implement a method to find the highest-paid employee and display their details.

CODE:



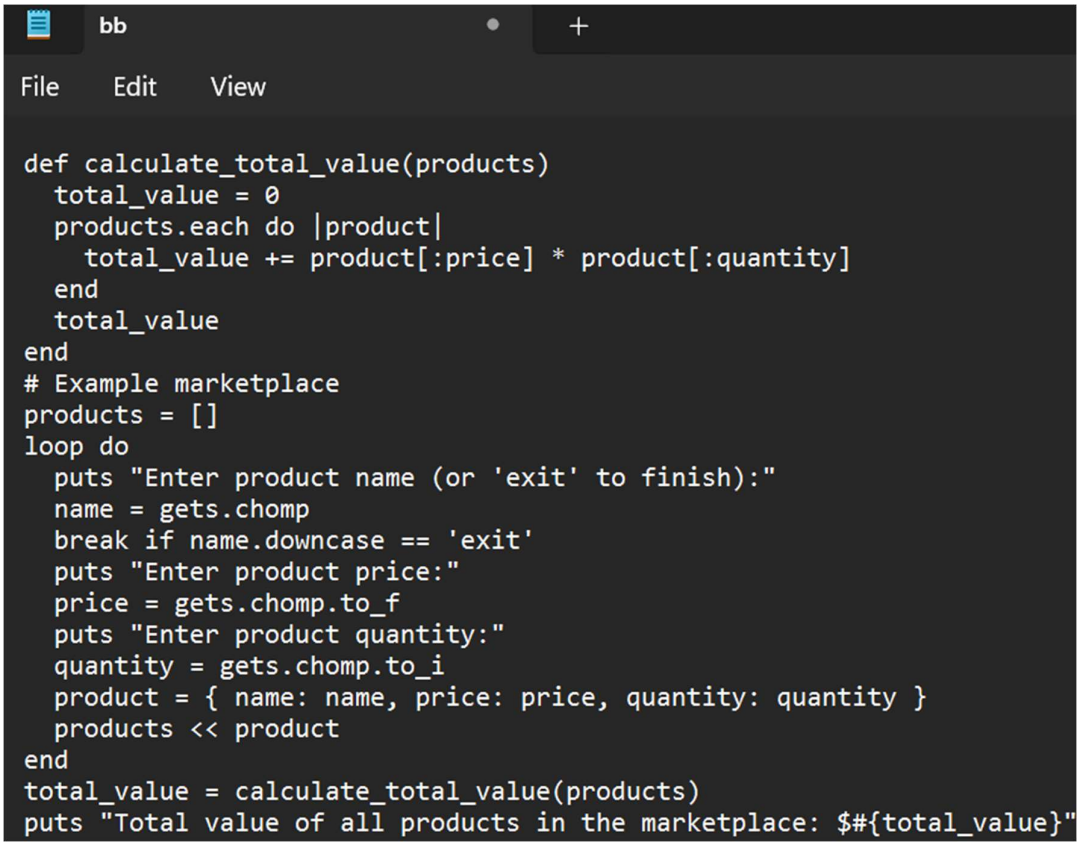
```
def find_highest_paid_employee(employees)
  highest_salary = 0
  highest_paid_employee = nil
  employees.each do |employee|
    if employee[:salary] > highest_salary
      highest_salary = employee[:salary]
      highest_paid_employee = employee
    end
  end

  highest_paid_employee
end

# Example employee database
employees = []
loop do

  puts "Enter employee name (or 'exit' to finish):"
  name = gets.chomp
  break if name.downcase == 'exit'
  puts "Enter employee department:"
  department = gets.chomp
  puts "Enter employee salary:"
  salary = gets.chomp.to_i
  employee = { name: name, department: department, salary: salary }
  employees << employee
end
highest_paid_employee = find_highest_paid_employee(employees)
if highest_paid_employee.nil?

  puts "No employees found."
Else

  puts "Highest-paid employee:"
  puts "Name: #{highest_paid_employee[:name]}"
  puts "Department: #{highest_paid_employee[:department]}"
  puts "Salary: #{highest_paid_employee[:salary]}"

end
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021 VIMAL KUMAR S>ruby bb.rb
Enter employee name (or 'exit' to finish):
Priyanka
Enter employee department:
Sales
Enter employee salary:
5000
Enter employee name (or 'exit' to finish):
Vimal
Enter employee department:
Service
Enter employee salary:
4000
Enter employee name (or 'exit' to finish):
exit
Highest-paid employee:
Name: Priyanka
Department: Sales
Salary: 5000
```

3. Scenario: Online Marketplace Question: You are creating an online marketplace where sellers can list their products. Design a hash structure to store information about products, including their names, prices, and quantities. Implement a method to calculate the total value of all products in the marketplace.

CODE:




```
def calculate_total_value(products)
 total_value = 0
 products.each do |product|
  total_value += product[:price] * product[:quantity]
 end
```

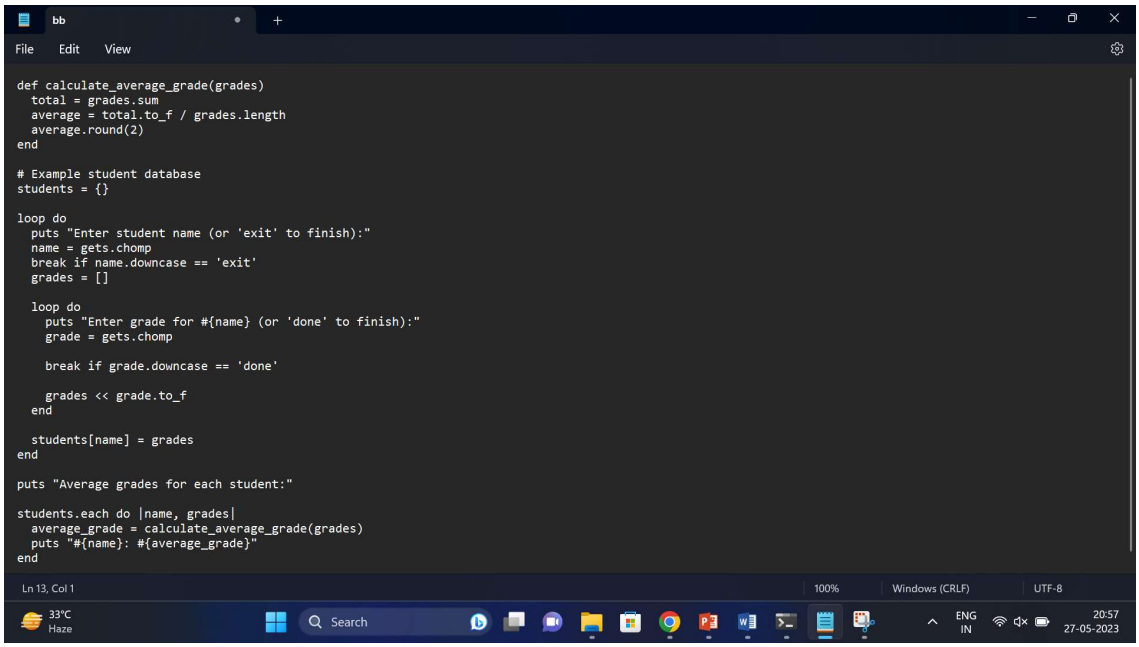
```
  total_value
end
# Example marketplace
products = []
loop do
  puts "Enter product name (or 'exit' to finish):"
  name = gets.chomp
  break if name.downcase == 'exit'
  puts "Enter product price:"
  price = gets.chomp.to_f
  puts "Enter product quantity:"
  quantity = gets.chomp.to_i
  product = { name: name, price: price, quantity: quantity }
  products << product
end
total_value = calculate_total_value(products)
puts "Total value of all products in the marketplace: $#{total_value}"
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021 VIMAL KUMAR S>ruby bb.rb
Enter product name (or 'exit' to finish):
Milk
Enter product price:
25
Enter product quantity:
5
Enter product name (or 'exit' to finish):
Boost
Enter product price:
50
Enter product quantity:
6
Enter product name (or 'exit' to finish):
Sugar
Enter product price:
99
Enter product quantity:
4
Enter product name (or 'exit' to finish):
exit
Total value of all products in the marketplace: $821.0

C:\Users\Dell\Desktop\21MIS1021 VIMAL KUMAR S>
```

4. Scenario: Student Grades Question: You are building a system to manage student grades. Design a hash structure to store information about multiple students, including their names and an array of grades for different subjects. Implement a method to calculate the average grade for each student and display the results.

   CODE:




```
def calculate_average_grade(grades)
  total = grades.sum
  average = total.to_f / grades.length
  average.round(2)
end
# Example student database
students = {}
loop do
  puts "Enter student name (or 'exit' to finish):"
  name = gets.chomp
  break if name.downcase == 'exit'
  grades = []
```

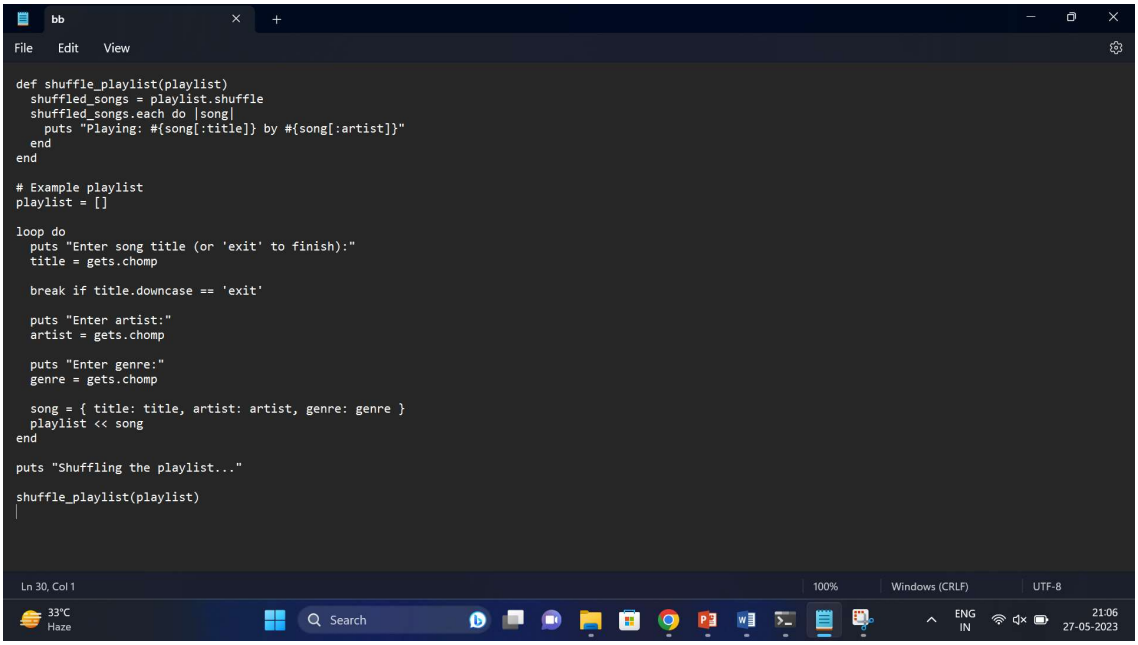
```
 loop do
    puts "Enter grade for #{name} (or 'done' to finish):"
    grade = gets.chomp
    break if grade.downcase == 'done'
    grades << grade.to_f
  end
  students[name] = grades
end
puts "Average grades for each student:"
students.each do |name, grades|
  average_grade = calculate_average_grade(grades)
  puts "#{name}: #{average_grade}"
end
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021 VIMAL KUMAR S>ruby bb.rb
Enter student name (or 'exit' to finish):
Vimal
Enter grade for Vimal (or 'done' to finish):
9
Enter grade for Vimal (or 'done' to finish):
10
Enter grade for Vimal (or 'done' to finish):
9
Enter grade for Vimal (or 'done' to finish):
done
Enter student name (or 'exit' to finish):
Priyanka
Enter grade for Priyanka (or 'done' to finish):
7
Enter grade for Priyanka (or 'done' to finish):
8
Enter grade for Priyanka (or 'done' to finish):
9
Enter grade for Priyanka (or 'done' to finish):
done
Enter student name (or 'exit' to finish):
exit
Average grades for each student:
Vimal: 9.33
Priyanka: 8.0
```

5. Scenario: Music Playlist Question: You are developing a music playlist application. Design a hash structure to store information about songs, including their titles, artists, and genres. Implement a method to shuffle the playlist randomly and play the songs in a random order.

CODE:




```
def shuffle_playlist(playlist)
  shuffled_songs = playlist.shuffle
  shuffled_songs.each do |song|
    puts "Playing: #{song[:title]} by #{song[:artist]}"
  end
end
# Example playlist
playlist = []
loop do
  puts "Enter song title (or 'exit' to finish):"
  title = gets.chomp
  break if title.downcase == 'exit'
```

```
  puts "Enter artist:"
  artist = gets.chomp
  puts "Enter genre:"
  genre = gets.chomp
  song = { title: title, artist: artist, genre: genre }
  playlist << song
end
puts "Shuffling the playlist..."
shuffle_playlist(playlist)
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021 VIMAL KUMAR S>ruby bb.rb
Enter song title (or 'exit' to finish):
Arabic Kuthu
Enter artist:
Anirudh
Enter genre:
Folk
Enter song title (or 'exit' to finish):
Kadhal Aasai
Enter artist:
Yuvan
Enter genre:
Love
Enter song title (or 'exit' to finish):
Vizhi Moodi Yosithal
Enter artist:
Harris Jeyaraj
Enter genre:
Love
Enter song title (or 'exit' to finish):
Siragugal
Enter artist:
Yuvan
Enter genre:
Love
Enter song title (or 'exit' to finish):
exit
Shuffling the playlist...
Playing: Vizhi Moodi Yosithal by Harris Jeyaraj
Playing: Siragugal by Yuvan
Playing: Kadhal Aasai by Yuvan
```