

Ethan Roberts

Ruby Assignment Output

CS 417

Roberts.E.assn05

Problem 1 (Source-code and output)

```
# Ethan Roberts
# CS 417 Topics in OOP
# This script will loop through an array and
# count how many times a "key" is found inside array
# Ruby Assignment

def occurrences(list,key)

  keyMatchCount = 0
  list.each do |num|
    if num == key
      keyMatchCount = keyMatchCount + 1
    end
  end

  return keyMatchCount
end

myAry = [10,20,30,0,50,20,20,70,80,80,72,90,100,90,20,100,100,100,50,50]
myKey = 94

print "Occurrences of key "
print myKey
print ": "
print occurrences(myAry,myKey)    #print how many occurrences of key
```

C:\Users\Ethan\Desktop\rubyProgram.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

rubyProgram.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3 # This script will loop through an array and
4 # count how many times a "key" is found inside array
5 # Ruby Assignment
6
7
8 def occurrences(list,key)
9
10   keyMatchCount = 0
11   list.each do |num|
12     if num == key
13       keyMatchCount = keyMatchCount + 1
14     end
15   end
16
17   return keyMatchCount
18 end
19
20
21 myAry = [10,20,30,0,50,20,20,70,80,80,90,100,90,20,100,100,100,50,50]
22 myKey = 20
23
24 print "Occurrences of key "
25 print myKey
26 print ": "
27 print occurrences(myAry,myKey) #print how many occurrences of key
28
29
```

Ruby file length: 548 lines: 29 Ln: 20 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Ethan\Desktop>ruby rubyProgram.rb
Occurrences of key 20: 4
C:\Users\Ethan\Desktop>
```

```
*C:\Users\Ethan\Desktop\rubyProgram.rb - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
rubyProgram.rb
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3 # This script will loop through an array and
4 # count how many times a "key" is found inside array
5 # Ruby Assignment
6
7
8 def occurrences(list,key)
9
10   keyMatchCount = 0
11   list.each do |num|
12     if num == key
13       keyMatchCount = keyMatchCount + 1
14     end
15   end
16
17   return keyMatchCount
18 end
19
20
21 myArray = [10,20,30,0,50,20,20,70,80,80,90,100,90,20,100,100,100,50,50]
22 myKey = 50
23
24 print "Occurrences of key "
25 print myKey
26 print ": "
27 print occurrences(myArray,myKey) #print how many occurrences of key
28
29

Ruby file      length: 548   lines: 29      Ln: 24   Col: 27   Sel: 0 | 0    Windows (CR LF)  UTF-8    INS
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Ethan\Desktop>ruby rubyProgram.rb
Occurrences of key 50: 3
C:\Users\Ethan\Desktop>
```

C:\Users\Ethan\Desktop\rubyProgram.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

rubyProgram.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3 # This script will loop through an array and
4 # count how many times a "key" is found inside array
5 # Ruby Assignment
6
7
8 def occurrences(list,key)
9
10   keyMatchCount = 0
11   list.each do |num|
12     if num == key
13       keyMatchCount = keyMatchCount + 1
14     end
15   end
16
17   return keyMatchCount
18 end
19
20
21 myArray = [10,20,30,0,50,20,20,70,80,80,90,100,90,20,100,100,100,50,50]
22 myKey = 100
23
24 print "Occurrences of key "
25 print myKey
26 print ": "
27 print occurrences(myArray,myKey) #print how many occurrences of key
28
29
```

Ruby file length: 549 lines: 29 Ln: 22 Col: 12 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Ethan\Desktop>ruby rubyProgram.rb
Occurrences of key 100: 4
C:\Users\Ethan\Desktop>
```

C:\Users\Ethan\Desktop\rubyProgram.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

rubyProgram.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3 # This script will loop through an array and
4 # count how many times a "key" is found inside array
5 # Ruby Assignment
6
7
8 def occurrences(list,key)
9
10   keyMatchCount = 0
11   list.each do |num|
12     if num == key
13       keyMatchCount = keyMatchCount + 1
14     end
15   end
16
17   return keyMatchCount
18 end
19
20
21 myAry = [10,20,30,0,50,20,20,70,80,80,72,90,100,90,20,100,100,100,50,50]
22 myKey = 72
23
24 print "Occurrences of key "
25 print myKey
26 print ": "
27 print occurrences(myAry,myKey) #print how many occurrences of key
28
29
```

Ruby file length: 551 lines: 29 Ln: 20 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Ethan\Desktop>ruby rubyProgram.rb
Occurrences of key 72: 1
C:\Users\Ethan\Desktop>
```

C:\Users\Ethan\Desktop\rubyProgram.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

rubyProgram.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3 # This script will loop through an array and
4 # count how many times a "key" is found inside array
5 # Ruby Assignment
6
7
8 def occurrences(list,key)
9
10     keyMatchCount = 0
11     list.each do |num|
12         if num == key
13             keyMatchCount = keyMatchCount + 1
14         end
15     end
16
17     return keyMatchCount
18 end
19
20
21 myArray = [10,20,30,0,50,20,20,70,80,80,72,90,100,90,20,100,100,100,50,50]
22 myKey = 94
23
24 print "Occurrences of key "
25 print myKey
26 print ": "
27 print occurrences(myArray,myKey) #print how many occurrences of key
28
29
```

Ruby file length: 551 lines: 29 Ln: 24 Col: 27 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
Occurrences of key 94: 0
C:\Users\Ethan\Desktop>
```

Problem 2 (Source-code and output)

```
# Ethan Roberts
# CS 417 Topics in OOP

# This script will loop through an array and return
# number of items that are less than item

# Ruby Assignment

def numSmaller(list,item)

  keyMatchCount = 0

  if list.length() == 0    #Making sure the array is not empty
    print "Array is empty, program terminated."
    return
  else
    list.each do |num|
      if num < item
        keyMatchCount = keyMatchCount + 1
      end
    end
  end

  return keyMatchCount
end

myAry = [0,1,2,3,4,5,6,7,8,9,10]
myItem = 8

print "Elements in array less than "
print myItem
print ": "
print numSmaller(myAry,myItem)
```

C:\Users\Ethan\Desktop\Ruby Assignment\problem2Script.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

problem2Script.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3
4 # This script will loop through an array and return
5 # number of items that are less than item
6
7 # Ruby Assignment
8
9
10 def numSmaller(list,item)
11
12     keyMatchCount = 0
13
14     if list.length() == 0 #Making sure the array is not empty
15         print "Array is empty, program terminated."
16         return
17     else
18         list.each do |num|
19             if num < item
20                 keyMatchCount = keyMatchCount + 1
21             end
22         end
23     end
24
25     return keyMatchCount
26 end
27
28
29
30 myArry = [0,1,1,1,2,2,3]
31 myItem = 4
32
33 print "Elements in array less than "
34 print myItem
35 print ": "
36 print numSmaller(myArry,myItem)
37
38
```

Ruby file length: 635 lines: 38 Ln: 38 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
Elements in array less than 4: 7
C:\Users\Ethan\Desktop\Ruby Assignment>
```


C:\Users\Ethan\Desktop\Ruby Assignment\problem2Script.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

problem2Script.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3
4 # This script will loop through an array and return
5 # number of items that are less than item
6
7 # Ruby Assignment
8
9
10 def numSmaller(list,item)
11
12     keyMatchCount = 0
13
14     if list.length() == 0 #Making sure the array is not empty
15         print "Array is empty, program terminated."
16         return
17     else
18         list.each do |num|
19             if num < item
20                 keyMatchCount = keyMatchCount + 1
21             end
22         end
23     end
24
25     return keyMatchCount
26 end
27
28
29
30 myAry = [0,1,1,1,2,2,3]
31 myItem = 2
32
33 print "Elements in array less than "
34 print myItem
35 print ": "
36 print numSmaller(myAry,myItem)
37
38
```

Ruby file length: 635 lines: 38 Ln: 31 Col: 11 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
Elements in array less than 2: 4
C:\Users\Ethan\Desktop\Ruby Assignment>
```

C:\Users\Ethan\Desktop\Ruby Assignment\problem2Script.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2

problem2Script.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3
4 # This script will loop through an array and return
5 # number of items that are less than item
6
7 # Ruby Assignment
8
9
10 def numSmaller(list,item)
11
12     keyMatchCount = 0
13
14     if list.length() == 0 #Making sure the array is not empty
15         print "Array is empty, program terminated."
16         return
17     else
18         list.each do |num|
19             if num < item
20                 keyMatchCount = keyMatchCount + 1
21             end
22         end
23     end
24
25     return keyMatchCount
26 end
27
28
29
30 myArray = []
31 myItem = 2
32
33 print "Elements in array less than "
34 print myItem
35 print ": "
36 print numSmaller(myArray,myItem)
37
38
```

Ruby file length: 622 lines: 38 Ln: 38 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Ethan\Desktop\Ruby Assignment>ruby problem2Script.rb
Elements in array less than 2: Array is empty, program terminated.
C:\Users\Ethan\Desktop\Ruby Assignment>
```

C:\Users\Ethan\Desktop\Ruby Assignment\problem2Script.rb - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2

problem2Script.rb

```
1 # Ethan Roberts
2 # CS 417 Topics in OOP
3
4 # This script will loop through an array and return
5 # number of items that are less than item
6
7 # Ruby Assignment
8
9
10 def numSmaller(list,item)
11
12     keyMatchCount = 0
13
14     if list.length() == 0 #Making sure the array is not empty
15         print "Array is empty, program terminated."
16         return
17     else
18         list.each do |num|
19             if num < item
20                 keyMatchCount = keyMatchCount + 1
21             end
22         end
23     end
24
25     return keyMatchCount
26 end
27
28
29
30 myAry = [0,1,2,3,4,5,6,7,8,9,10]
31 myItem = 8
32
33 print "Elements in array less than "
34 print myItem
35 print ": "
36 print numSmaller(myAry,myItem)
37
38
```

Ruby file length: 644 lines: 38 Ln: 38 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Ethan\Desktop\Ruby Assignment>ruby problem2Script.rb
Elements in array less than 8: 8
C:\Users\Ethan\Desktop\Ruby Assignment>
```

Problem 3 (Source-code and output)

```
# Ethan Roberts
# CS 417 Topics in OOP
# This script will find the second smallest index position.

# Ruby Assignment
```

```
# method for looping through array
# and seeing if smallest value has duplicate.
# If so, the second duplicate is second smallest
```

```
def scanSmallestDuplicate(smallestValue,list)
```

```
  x = 0
  dupCounter = 0
  indexLocator = 0
  while x < list.length && dupCounter < 2
    if list[x] == smallestValue
      indexLocator = x
      dupCounter = dupCounter + 1
    end
    x = x + 1
  end
```

```
  if dupCounter > 1
    return indexLocator
  else
    return 0
  end
end
```

```
def secondSmallest(list)
```

```
  smallest = list[0]
  secondSmallest = list[1]
  temp = 0
  secondSmallestIndex = 0
```

```
  # Compare 0th element and 1st element
  # in array
```

```
  if secondSmallest < smallest
    temp = smallest
    smallest = secondSmallest
    secondSmallest = temp
    secondSmallestIndex = 1 #referencing index 1 of array
  end
```

```
  i = 2 # index 0 and 1 already compared
```

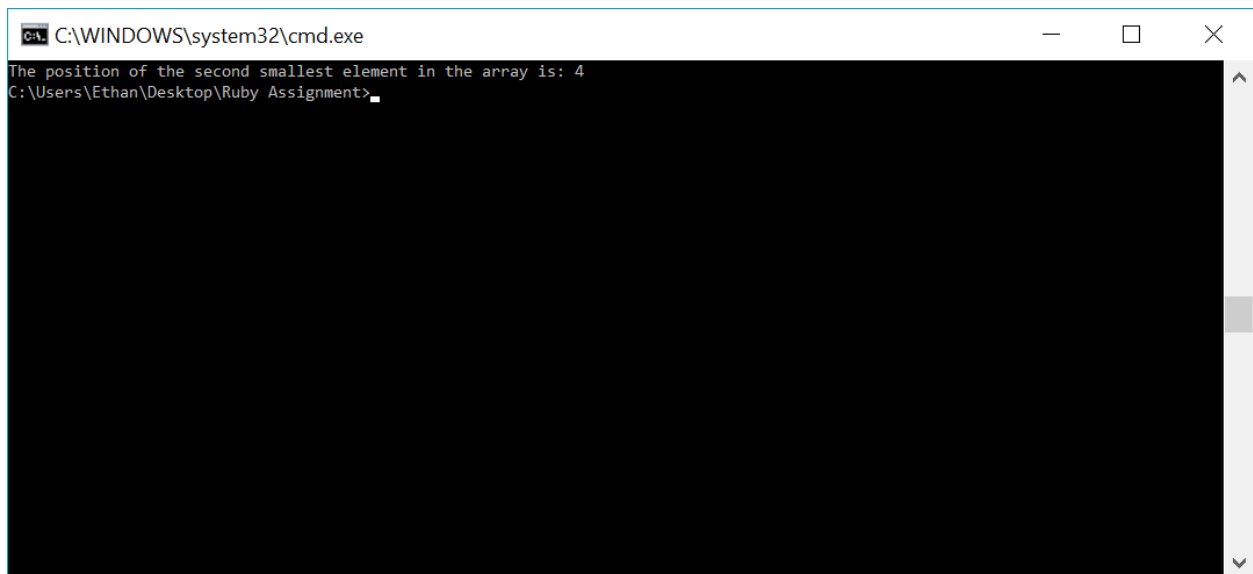
```
  while i < list.length
    if list[i] < secondSmallest
```

```
    if list[i] < smallest
      temp = smallest
      smallest = list[i]
      secondSmallest = temp
    else
      secondSmallest = list[i]
    end
  end
  i = i + 1
end

smallDupIndexLocator = scanSmallestDuplicate(smallest, list)
if smallDupIndexLocator > 0
  return smallDupIndexLocator
else
  return list.index(secondSmallest)
end
end

myAry = [9, 8, 6, 22, 4, 6, 77, 4]

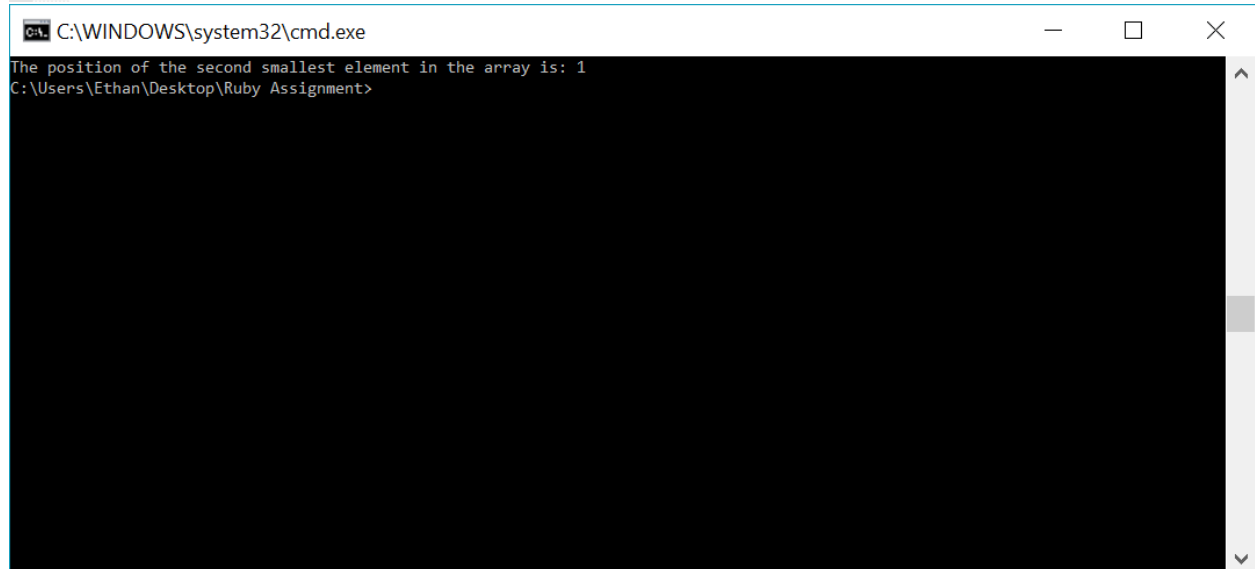
print "The position of the second smallest element in the array is: "
print secondSmallest(myAry)
```



A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The window contains the output of a Ruby script: "The position of the second smallest element in the array is: 4". Below this, the prompt "C:\Users\Ethan\Desktop\Ruby Assignment>" is visible, followed by a cursor. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
myAry = [0,1,2,3,4,5,6,7,8,9,10]
```

```
print "The position of the second smallest element in the array is: "  
print secondSmallest(myAry)
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt displays the output of a Ruby script: "The position of the second smallest element in the array is: 1". Below the output, the prompt "C:\Users\Ethan\Desktop\Ruby Assignment>" is visible. The command prompt area is mostly black, indicating that the rest of the script's output or the user's input is not visible in this view.

```
myAry = [10,9,8,7,6,5,4,3,2,1]
```

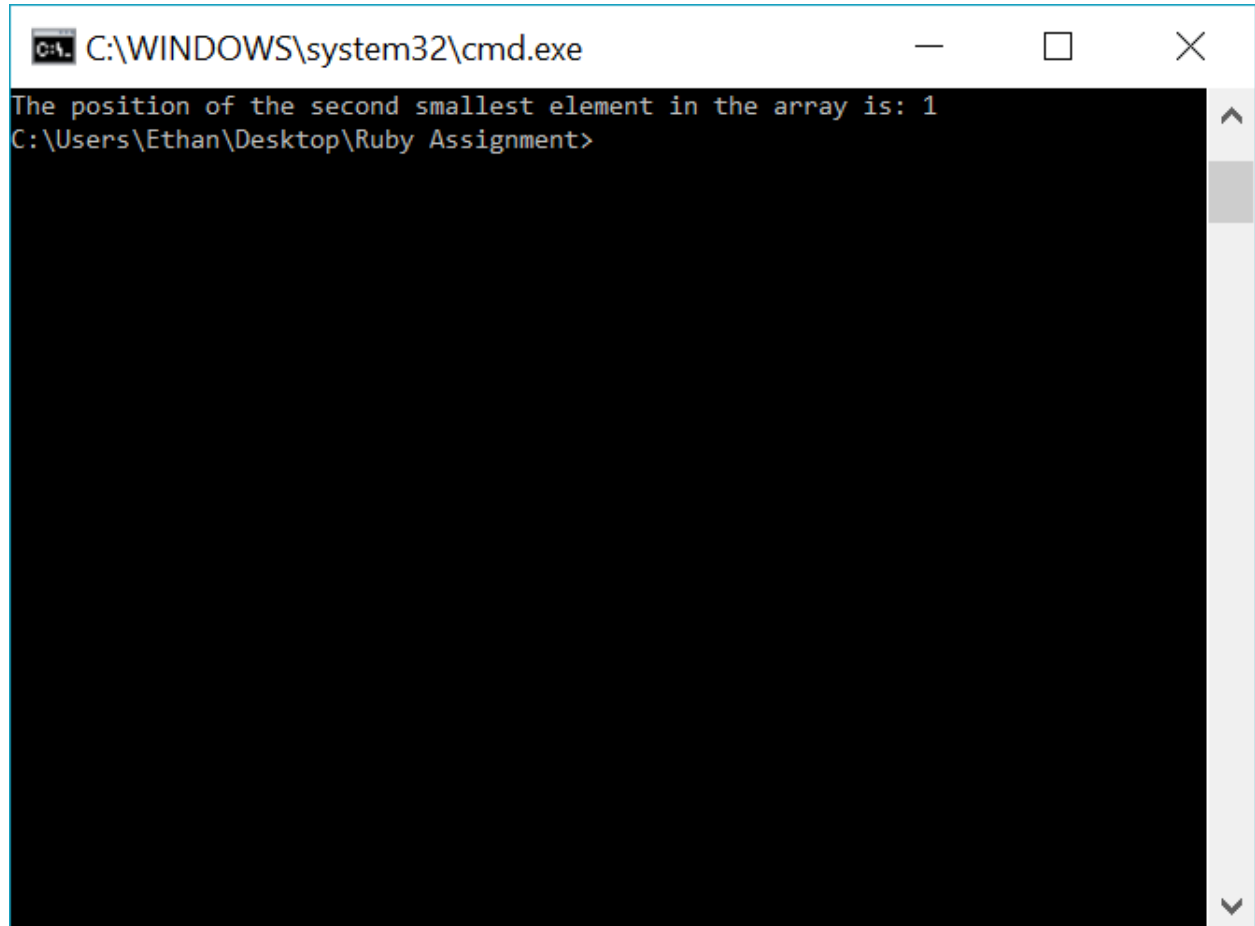
```
print "The position of the second smallest element in the array is: "  
print secondSmallest(myAry)
```

C:\WINDOWS\system32\cmd.exe

```
The position of the second smallest element in the array is: 8  
C:\Users\Ethan\Desktop\Ruby Assignment>
```

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

```
print "The position of the second smallest element in the array is: "  
print secondSmallest(myAry)
```



A screenshot of a Windows command prompt window. The title bar shows the path `C:\WINDOWS\system32\cmd.exe`. The window has standard minimize, maximize, and close buttons. The command prompt area is black with white text. It displays the output of a Ruby program: `The position of the second smallest element in the array is: 1`. Below this, the prompt `C:\Users\Ethan\Desktop\Ruby Assignment>` is visible. A vertical scrollbar is on the right side of the command prompt area.

```
C:\WINDOWS\system32\cmd.exe  
The position of the second smallest element in the array is: 1  
C:\Users\Ethan\Desktop\Ruby Assignment>
```



```
myAry = [9,8,6,22,4,6,77,4]
```

```
print "The position of the second smallest element in the array is: "
```

```
print secondSmallest(myAry)
```

C:\WINDOWS\system32\cmd.exe

The position of the second smallest element in the array is: 7

C:\Users\Ethan\Desktop\Ruby Assignment>

Problem 4 (Source-code and output)

```
# Ethan Roberts
# CS 417 Topics in OOP

# This script will contain two different
# methods for iterating over an array

# Ruby Assignment

class Array

  #NOTE: Used source:
  "https://stackoverflow.com/questions/16422872/reimplementing-enumerable-map-
  method-in-ruby"
  # for help understanding how passing a block works and for syntax help

  def iterativeMap
    out = []
    if block_given?
      self.each { |n| out << yield(n)}
      return out
    end
  end

  def recursiveMap
    if self.empty?
      return
    else
      number = self.pop
      number = number * 2
      recursiveMap
    end

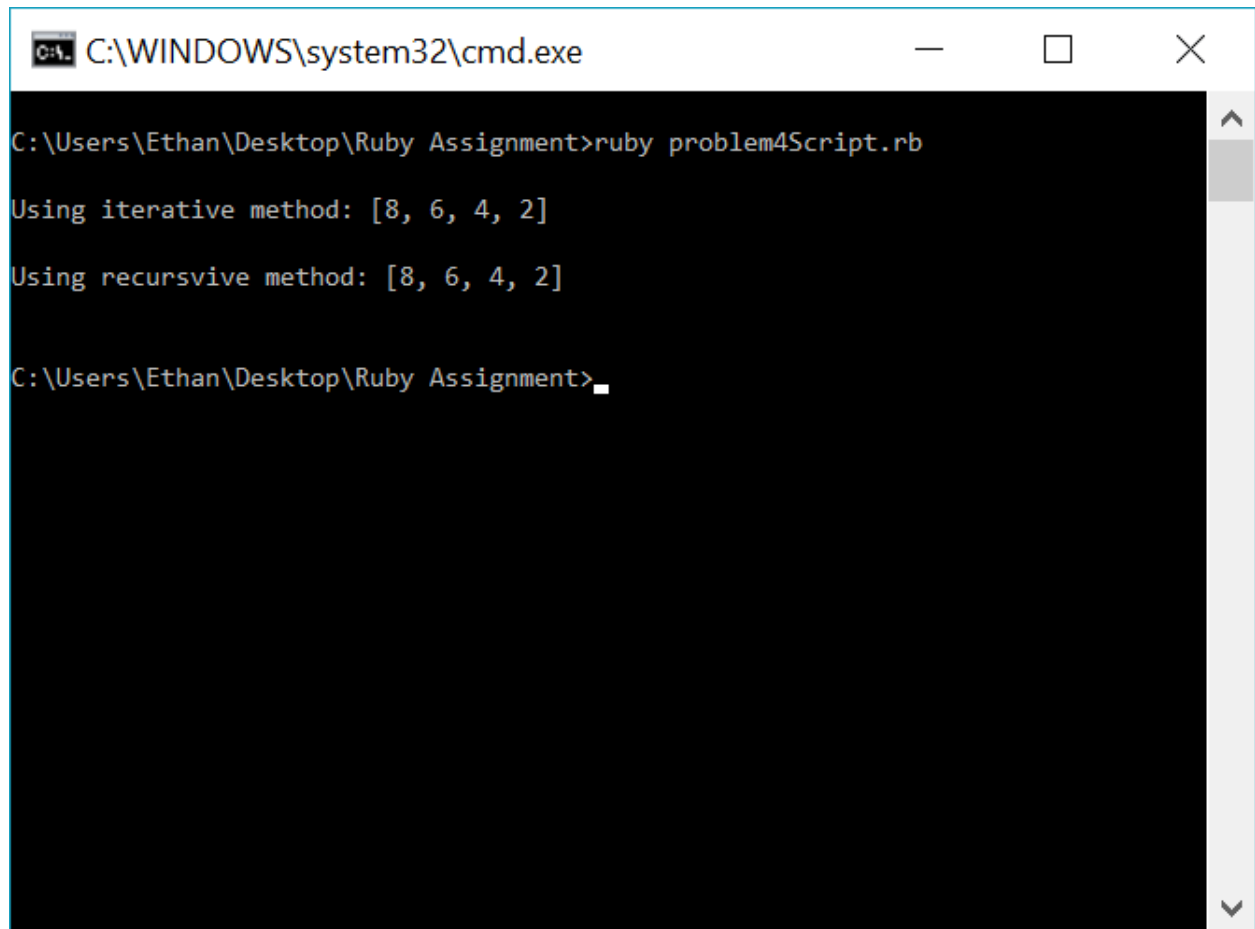
    return self << number #returning array with calculated values
  end
end

answerArray = [] # for holding final answer

#Section for testing "iterativeMap" method
myAry = [4,3,2,1]
answerArray = myAry.iterativeMap{|n| n * 2}
print "\nUsing iterative method: "
print answerArray
print "\n\n"

#Section for testing "recursiveMap" method
answerArray = myAry.recursiveMap {|n| n * 2}
print "Using recursive method: "
print answerArray
```

```
print "\n\n"
```



A screenshot of a Windows Command Prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe" and includes standard minimize, maximize, and close buttons. The command prompt shows the following text:

```
C:\Users\Ethan\Desktop\Ruby Assignment>ruby problem4Script.rb  
Using iterative method: [8, 6, 4, 2]  
Using recursive method: [8, 6, 4, 2]  
C:\Users\Ethan\Desktop\Ruby Assignment>
```

The output of the script shows two methods, iterative and recursive, both returning the array [8, 6, 4, 2]. A vertical scrollbar is visible on the right side of the command prompt window.

```
#Section for testing "iterativeMap" method
```

```
myAry = [50,40,30,20,10]
```

```
answerAry = myAry.iterativeMap{|n| n * 2}
```

```
print "\nUsing iterative method: "
```

```
print answerAry
```

```
print "\n\n"
```

```
#Section for testing "recursiveMap" method
```

```
answerAry = myAry.recursiveMap {|n| n * 2}
```

```
print "Using recursive method: "
```

```
print answerAry
```

```
print "\n\n"
```

C:\WINDOWS\system32\cmd.exe

```
Using iterative method: [100, 80, 60, 40, 20]
```

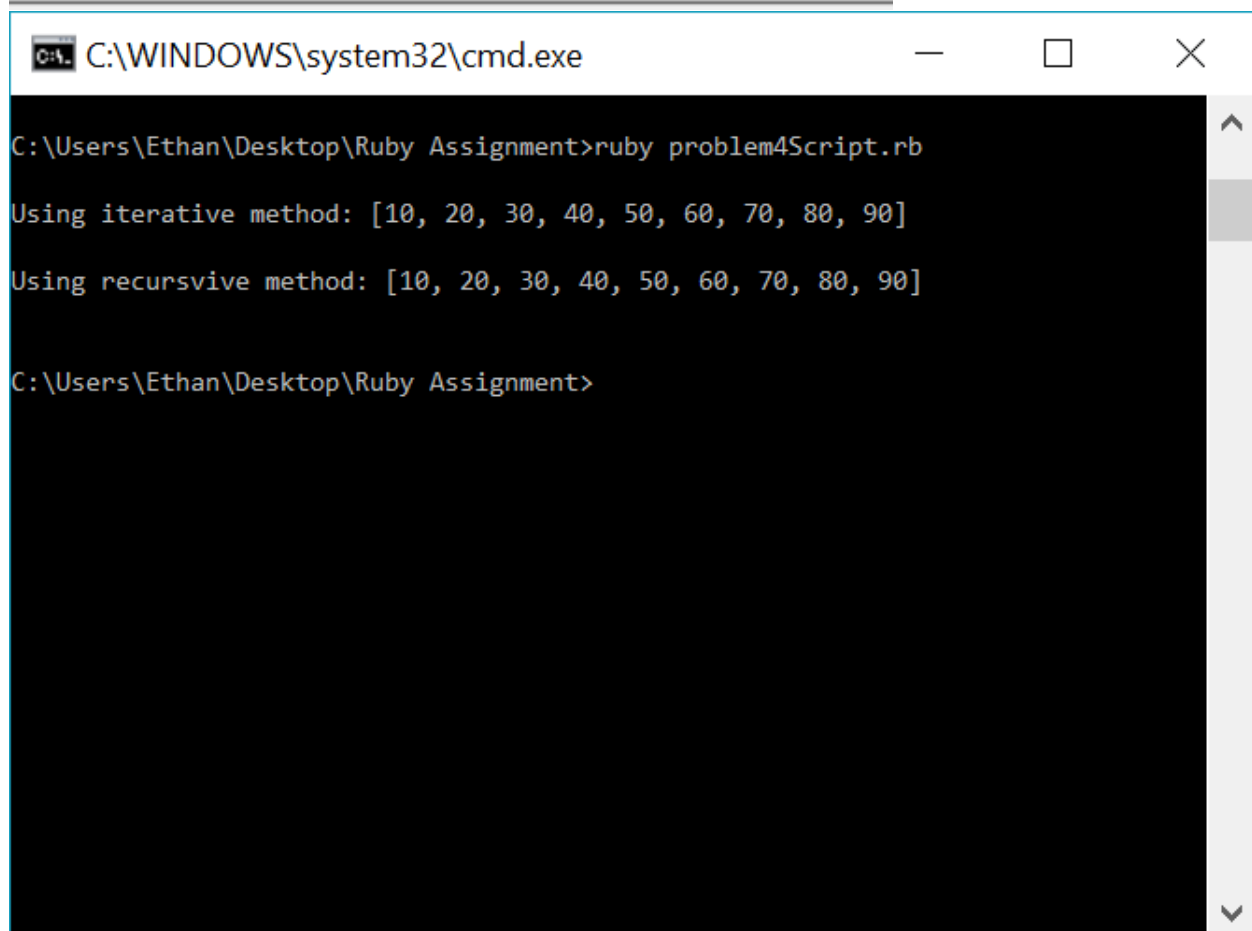
```
Using recursive method: [100, 80, 60, 40, 20]
```

```
C:\Users\Ethan\Desktop\Ruby Assignment>
```

```
answerAry = [] # for holding final answer

#Section for testing "iterativeMap" method
myAry = [5,10,15,20,25,30,35,40,45]
answerAry = myAry.iterativeMap{|n| n * 2}
print "\nUsing iterative method: "
print answerAry
print "\n\n"

#Section for testing "recursiveMap" method
answerAry = myAry.recursiveMap {|n| n * 2}
print "Using recursive method: "
print answerAry
print "\n\n"
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt is at "C:\Users\Ethan\Desktop\Ruby Assignment>". The user has entered the command "ruby problem4Script.rb". The output of the script is displayed on two lines: "Using iterative method: [10, 20, 30, 40, 50, 60, 70, 80, 90]" and "Using recursive method: [10, 20, 30, 40, 50, 60, 70, 80, 90]". The prompt is now at "C:\Users\Ethan\Desktop\Ruby Assignment>".

```
C:\WINDOWS\system32\cmd.exe

C:\Users\Ethan\Desktop\Ruby Assignment>ruby problem4Script.rb
Using iterative method: [10, 20, 30, 40, 50, 60, 70, 80, 90]
Using recursive method: [10, 20, 30, 40, 50, 60, 70, 80, 90]

C:\Users\Ethan\Desktop\Ruby Assignment>
```

Problem 5 (Source-code and output)

```
# Ethan Roberts
# CS 417 Topics in OOP

# This script will manipulates a list
# Ruby Assignment

# To understand instance variables for classes,
# I used: http://ruby-for-beginners.rubymonstas.org/writing\_classes/instance\_variables.html

class List

  def initialize # self-note: initialize is a ruby-defined "constructor"
    @backingStore = [] # @ sign means instance variable
  end

  def insert(value)
    @backingStore.push(value)
  end

  # cite: https://ruby-doc.org/core-2.2.0/Array.html#method-i-index
  # implemented ".index" method from this site
  def delete(value)
    if @backingStore.length() == 0 # backingStore is empty
      return
    else
      @backingStore.each do |n|
        if n == value
          @backingStore[@backingStore.index(n)] = nil
        end
      end
    end
  end

  def traverse
    @backingStore.each do |n|
      print n
      print " "
    end
  end

end
```

```

#----- BEGIN TESTING CODE -----

myList = List.new
stringList = List.new

myList.insert(1)
myList.insert(2)
myList.insert(3)
myList.insert(4)
myList.insert(5)
myList.insert(6)
myList.insert(7)
myList.insert(8)
print "Printing full list before deletes: "
myList.traverse
print "\n"
myList.delete(4)
print "After deletion: "
myList.traverse
print "\n"
myList.delete(3)
myList.delete(7)
print "After deletion: "
myList.traverse
print "\n"
myList.delete(1)
print "After deletion: "
myList.traverse
print "\n"
myList.insert(988)
print "After insert: "
myList.traverse
print "\n\n**Begin testing different data type...\n\n"

#----- TESTING DIFFERENT OBJECT DATA-TYPE -----

stringList.insert("cat")
stringList.insert("dog")
stringList.insert("shark")
stringList.insert("parrot")
stringList.insert("lizard")
stringList.insert("tiger")
stringList.insert("pelican")
print "Printing full list before deletes: "
stringList.traverse
print "\n"
stringList.delete("pelican")
print "After deletion: "
stringList.traverse
print "\n"
stringList.delete("shark")
print "After deletion: "
stringList.traverse
print "\n"

```

C:\WINDOWS\system32\cmd.exe

Printing full list before deletes: 1 2 3 4 5 6 7 8

After deletion: 1 2 3 5 6 7 8

After deletion: 1 2 5 6 8

After deletion: 2 5 6 8

After insert: 2 5 6 8 988

**Begin testing different data type...

Printing full list before deletes: cat dog shark parrot lizard tiger pelican

After deletion: cat dog shark parrot lizard tiger

After deletion: cat dog parrot lizard tiger

C:\Users\Ethan\Desktop\Ruby Assignment>