Problem Statement

HATFD1030

Rotate an Array Right by K Positions

Write a program to rotate an array right by k positions without using any built-in array or rotation

functions. For example, rotating [1, 2, 3, 4, 5] by 2 would give [4, 5, 1, 2, 3].

Instructions: You should implement the logic manually for rotating the array.

Program Code

```
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end
```

```
# Example 1:
arr = [1, 2, 3, 4, 5]
k = 1
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"
```

Output:

```
Rotated Array: [5, 1, 2, 3, 4]
```

```
# Example 2:
arr = [10, 20, 30, 40]
k = 2
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"
```

Output:

```
Rotated Array: [30, 40, 10, 20]
```

```
# Example 3:
arr = [7, 8, 9, 10, 11, 12]
k = 3
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"
```

Output:

```
Rotated Array: [10, 11, 12, 7, 8, 9]
```

# Example 4:

arr = [3, 6, 9]

k = 4

rotated_array = rotate_array(arr, k)

puts "Rotated Array: #{rotated_array}"

Output:

Rotated Array: [9, 3, 6]

# Example 5:

arr = [100, 200, 300, 400, 500, 600]

k = 5

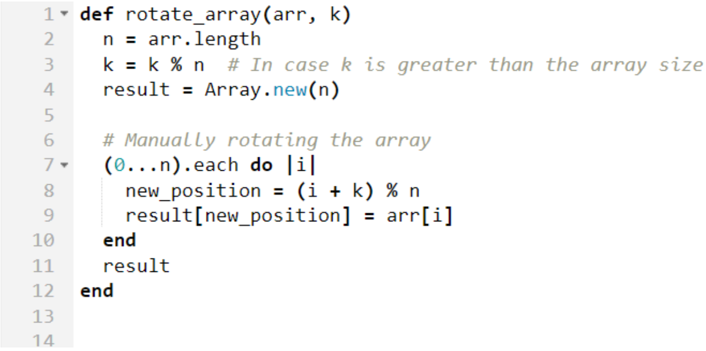
rotated_array = rotate_array(arr, k)

puts "Rotated Array: #{rotated_array}"

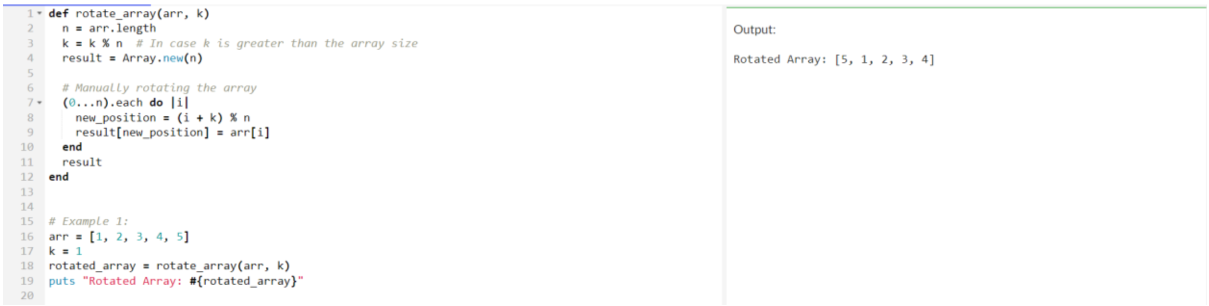
Output:

Rotated Array: [200, 300, 400, 500, 600, 100]

Step by Step Screenshots



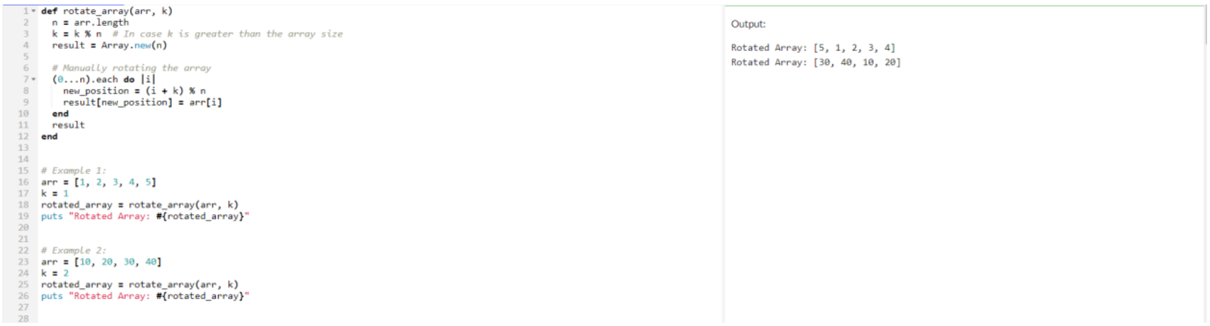
```
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end


```



```ruby
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end


# Example 1:
arr = [1, 2, 3, 4, 5]
k = 1
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"
```

Output:

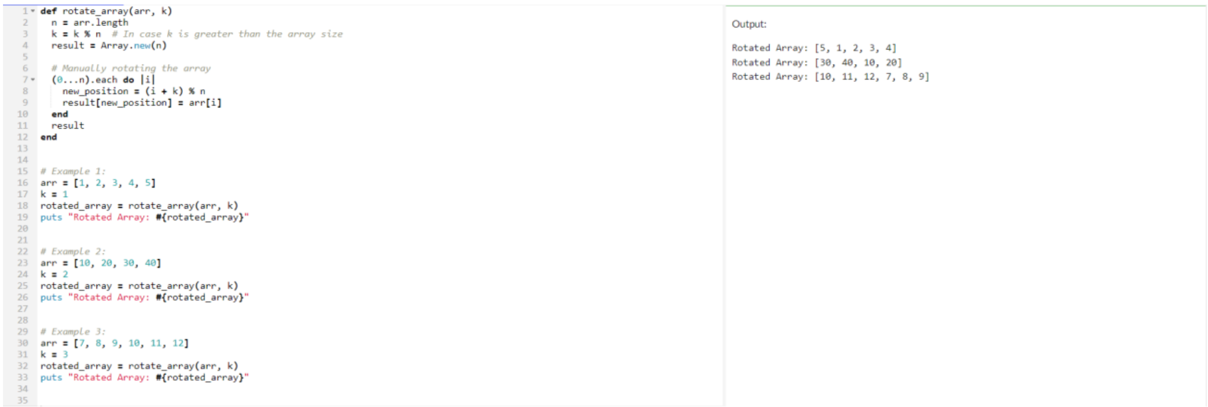
```
Rotated Array: [5, 1, 2, 3, 4]
```



```ruby
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end


# Example 1:
arr = [1, 2, 3, 4, 5]
k = 1
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 2:
arr = [10, 20, 30, 40]
k = 2
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"
```

Output:

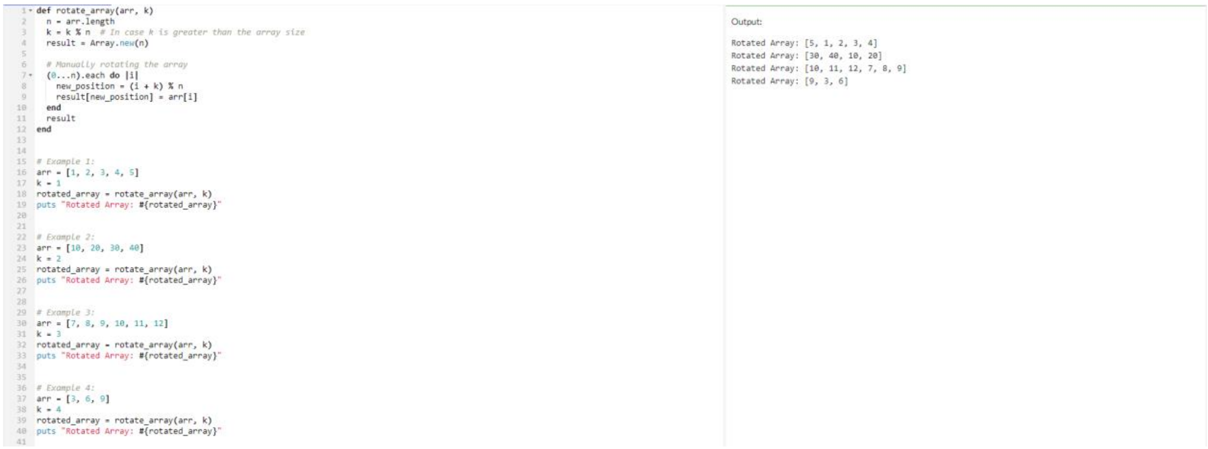
```
Rotated Array: [5, 1, 2, 3, 4]
Rotated Array: [30, 40, 10, 20]
```



```ruby
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end


# Example 1:
arr = [1, 2, 3, 4, 5]
k = 1
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 2:
arr = [10, 20, 30, 40]
k = 2
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 3:
arr = [7, 8, 9, 10, 11, 12]
k = 3
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"
```

Output:

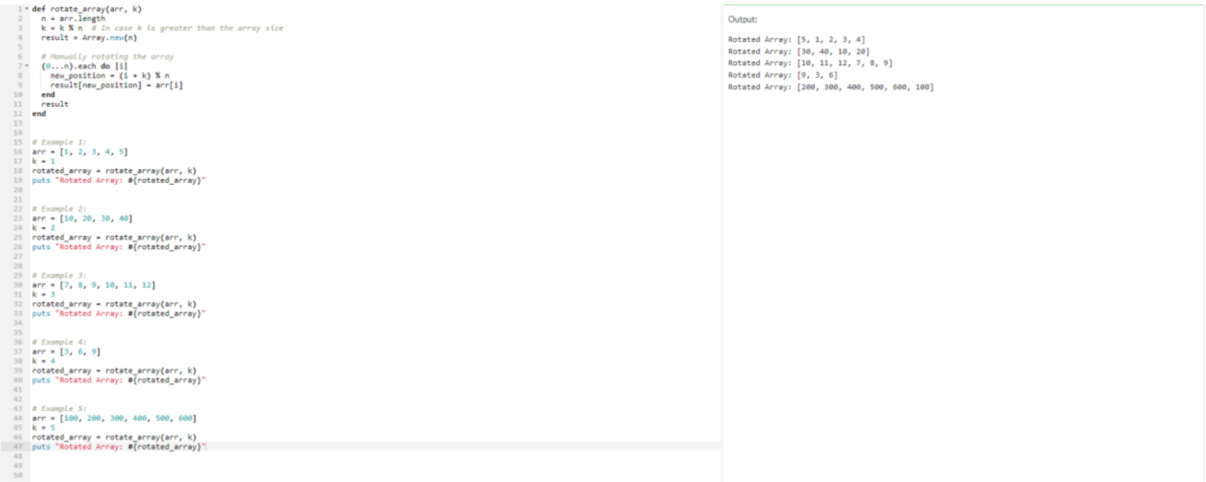
```
Rotated Array: [5, 1, 2, 3, 4]
Rotated Array: [30, 40, 10, 20]
Rotated Array: [10, 11, 12, 7, 8, 9]
```



```ruby
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end


# Example 1:
arr = [1, 2, 3, 4, 5]
k = 1
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 2:
arr = [10, 20, 30, 40]
k = 2
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 3:
arr = [7, 8, 9, 10, 11, 12]
k = 3
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 4:
arr = [3, 6, 9]
k = 4
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"

```

Output:

```
Rotated Array: [5, 1, 2, 3, 4]
Rotated Array: [30, 40, 10, 20]
Rotated Array: [10, 11, 12, 7, 8, 9]
Rotated Array: [9, 3, 6]
```



```ruby
def rotate_array(arr, k)
  n = arr.length
  k = k % n  # In case k is greater than the array size
  result = Array.new(n)

  # Manually rotating the array
  (0...n).each do |i|
    new_position = (i + k) % n
    result[new_position] = arr[i]
  end
  result
end


# Example 1:
arr = [1, 2, 3, 4, 5]
k = 1
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 2:
arr = [10, 20, 30, 40]
k = 2
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 3:
arr = [7, 8, 9, 10, 11, 12]
k = 3
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 4:
arr = [3, 6, 9]
k = 4
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"


# Example 5:
arr = [100, 200, 300, 400, 500, 600]
k = 5
rotated_array = rotate_array(arr, k)
puts "Rotated Array: #{rotated_array}"



```

Output:

```
Rotated Array: [5, 1, 2, 3, 4]
Rotated Array: [30, 40, 10, 20]
Rotated Array: [10, 11, 12, 7, 8, 9]
Rotated Array: [9, 3, 6]
Rotated Array: [200, 300, 400, 500, 600, 100]
```