

## 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

```
1 def rotate_array(arr, k)
2   n = arr.length
3   k = k % n # In case k is greater than the array size
4   result = Array.new(n)
5
6   # Manually rotating the array
7   (0...n).each do |i|
8     new_position = (i + k) % n
9     result[new_position] = arr[i]
10  end
11  result
12 end
13
14
```

```

1 def rotate_array(arr, k)
2   n = arr.length
3   k = k % n # In case k is greater than the array size
4   result = Array.new(n)
5
6   # Manually rotating the array
7   (0..n).each do |i|
8     new_position = (i + k) % n
9     result[new_position] = arr[i]
10  end
11  result
12 end
13
14 # Example 1:
15 arr = [1, 2, 3, 4, 5]
16 k = 1
17 rotated_array = rotate_array(arr, k)
18 puts "Rotated Array: #{rotated_array}"
19
20

```

Output:

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

```

1 def rotate_array(arr, k)
2   n = arr.length
3   k = k % n # In case k is greater than the array size
4   result = Array.new(n)
5
6   # Manually rotating the array
7   (0..n).each do |i|
8     new_position = (i + k) % n
9     result[new_position] = arr[i]
10  end
11  result
12 end
13
14 # Example 1:
15 arr = [1, 2, 3, 4, 5]
16 k = 1
17 rotated_array = rotate_array(arr, k)
18 puts "Rotated Array: #{rotated_array}"
19
20 # Example 2:
21 arr = [10, 20, 30, 40]
22 k = 2
23 rotated_array = rotate_array(arr, k)
24 puts "Rotated Array: #{rotated_array}"
25
26
27
28

```

Output:

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

```

1 def rotate_array(arr, k)
2   n = arr.length
3   k = k % n # In case k is greater than the array size
4   result = Array.new(n)
5
6   # Manually rotating the array
7   (0..n).each do |i|
8     new_position = (i + k) % n
9     result[new_position] = arr[i]
10  end
11  result
12 end
13
14 # Example 1:
15 arr = [1, 2, 3, 4, 5]
16 k = 1
17 rotated_array = rotate_array(arr, k)
18 puts "Rotated Array: #{rotated_array}"
19
20 # Example 2:
21 arr = [10, 20, 30, 40]
22 k = 2
23 rotated_array = rotate_array(arr, k)
24 puts "Rotated Array: #{rotated_array}"
25
26 # Example 3:
27 arr = [7, 8, 9, 10, 11, 12]
28 k = 3
29 rotated_array = rotate_array(arr, k)
30 puts "Rotated Array: #{rotated_array}"
31
32
33
34
35

```

Output:

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

```

1 def rotate_array(arr, k)
2   n = arr.length
3   k = k % n # In case k is greater than the array size
4   result = Array.new(n)
5
6   # Manually rotating the array
7   (0..n).each do |i|
8     new_position = (i + k) % n
9     result[new_position] = arr[i]
10  end
11  result
12 end
13
14
15 # Example 1:
16 arr = [1, 2, 3, 4, 5]
17 k = 1
18 rotated_array = rotate_array(arr, k)
19 puts "Rotated Array: #{rotated_array}"
20
21
22 # Example 2:
23 arr = [10, 20, 30, 40]
24 k = 2
25 rotated_array = rotate_array(arr, k)
26 puts "Rotated Array: #{rotated_array}"
27
28
29 # Example 3:
30 arr = [7, 8, 9, 10, 11, 12]
31 k = 3
32 rotated_array = rotate_array(arr, k)
33 puts "Rotated Array: #{rotated_array}"
34
35
36 # Example 4:
37 arr = [3, 6, 9]
38 k = 4
39 rotated_array = rotate_array(arr, k)
40 puts "Rotated Array: #{rotated_array}"
41

```

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]

```

```

1 def rotate_array(arr, k)
2   n = arr.length
3   k = k % n # In case k is greater than the array size
4   result = Array.new(n)
5
6   # Manually rotating the array
7   (0..n).each do |i|
8     new_position = (i + k) % n
9     result[new_position] = arr[i]
10  end
11  result
12 end
13
14
15 # Example 1:
16 arr = [1, 2, 3, 4, 5]
17 k = 1
18 rotated_array = rotate_array(arr, k)
19 puts "Rotated Array: #{rotated_array}"
20
21
22 # Example 2:
23 arr = [10, 20, 30, 40]
24 k = 2
25 rotated_array = rotate_array(arr, k)
26 puts "Rotated Array: #{rotated_array}"
27
28
29 # Example 3:
30 arr = [7, 8, 9, 10, 11, 12]
31 k = 3
32 rotated_array = rotate_array(arr, k)
33 puts "Rotated Array: #{rotated_array}"
34
35
36 # Example 4:
37 arr = [3, 6, 9]
38 k = 4
39 rotated_array = rotate_array(arr, k)
40 puts "Rotated Array: #{rotated_array}"
41
42
43 # Example 5:
44 arr = [100, 200, 300, 400, 500, 600]
45 k = 5
46 rotated_array = rotate_array(arr, k)
47 puts "Rotated Array: #{rotated_array}"
48
49
50

```

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]

```