

## HATFD1030

### Problem Statement:

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.

### Solution Code:

```
#!/bin/bash

rotate_array() {
    local arr=("$@")
    local n=${#arr[@]}
    local k=$1
    shift

    k=$((k % n))
    if [ $k -eq 0 ]; then
        echo "${arr[@]}"
        return
    fi

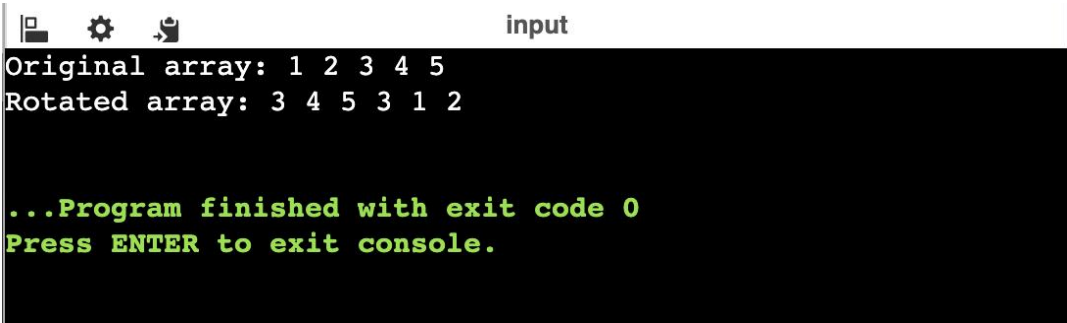
    local temp=("${arr[@]: -k}")
    temp+=("${arr[@]:0:n-k}")

    echo "${temp[@]}"
}

array=(1 2 3 4 5)
k=2

echo "Original array: ${array[@]}"
echo "Rotated array: $(rotate_array $k "${array[@]}")"
```

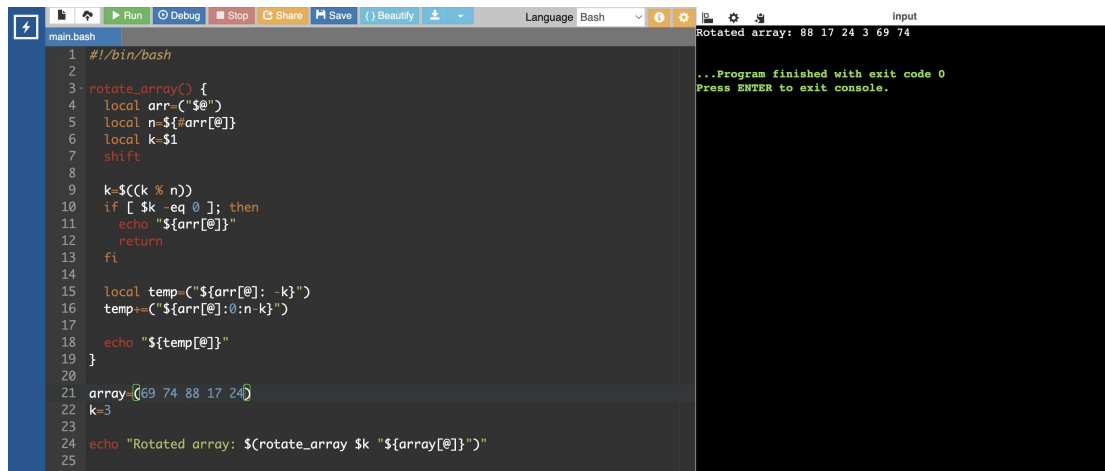
### Output:

A terminal window titled 'input' with a dark background. It shows the output of the script: 'Original array: 1 2 3 4 5' and 'Rotated array: 3 4 5 3 1 2'. At the bottom, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.' in green text.

```
input
Original array: 1 2 3 4 5
Rotated array: 3 4 5 3 1 2

...Program finished with exit code 0
Press ENTER to exit console.
```

## Sample Output 1:

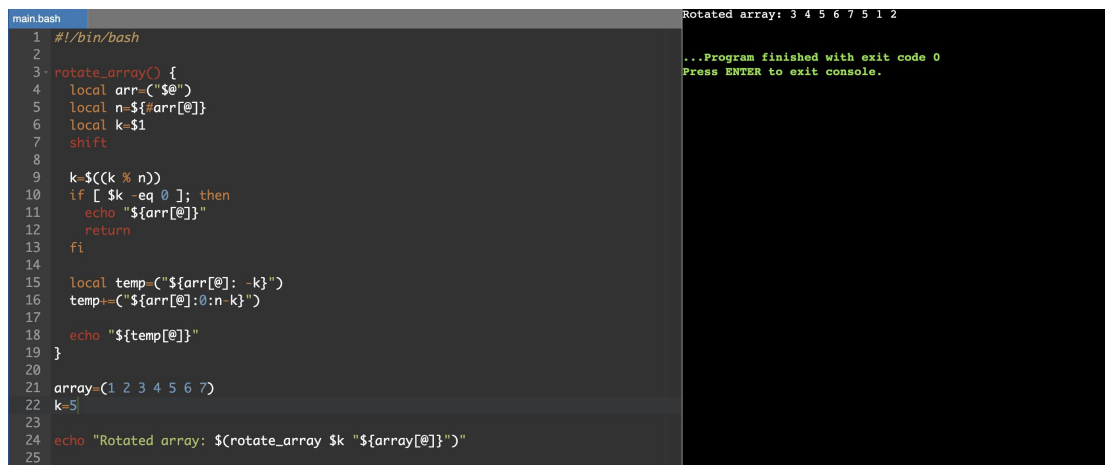


```
1 #!/bin/bash
2
3 rotate_array() {
4     local arr=("$@")
5     local n=${#arr[@]}
6     local k=$1
7     shift
8
9     k=$((k % n))
10    if [ $k -eq 0 ]; then
11        echo "${arr[@]}"
12        return
13    fi
14
15    local temp=("${arr[@]: -$k}")
16    temp+=("${arr[@]:0:n-k}")
17
18    echo "${temp[@]}"
19 }
20
21 array=(69 74 88 17 24 3)
22 k=3
23
24 echo "Rotated array: $(rotate_array $k "${array[@]}")"
```

Rotated array: 88 17 24 3 69 74

...Program finished with exit code 0  
Press ENTER to exit console.

## Sample Output 2:



```
1 #!/bin/bash
2
3 rotate_array() {
4     local arr=("$@")
5     local n=${#arr[@]}
6     local k=$1
7     shift
8
9     k=$((k % n))
10    if [ $k -eq 0 ]; then
11        echo "${arr[@]}"
12        return
13    fi
14
15    local temp=("${arr[@]: -$k}")
16    temp+=("${arr[@]:0:n-k}")
17
18    echo "${temp[@]}"
19 }
20
21 array=(1 2 3 4 5 6 7)
22 k=5
23
24 echo "Rotated array: $(rotate_array $k "${array[@]}")"
```

Rotated array: 3 4 5 6 7 5 1 2

...Program finished with exit code 0  
Press ENTER to exit console.