# Search in Rotated Sorted Array

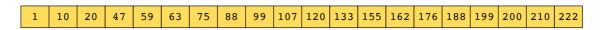
Try to solve the Search in Rotated Sorted Array problem.



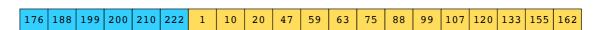
### **Statement**

Given a sorted integer array, nums, and an integer value, target, the array is rotated by some arbitrary number. Search and return the index of target in this array. If the target does not exist, return -1.

An original sorted array before rotation is given below:



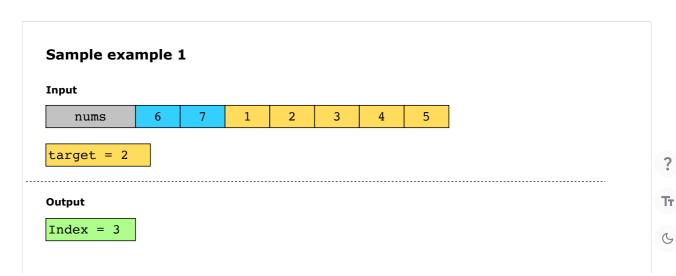
After rotating this array 6 times, it changes to:



### Constraints

- All values in nums are unique.
- The values in nums are in sorted in ascending order.
- The array may have been rotated by some arbitrary number.
- $1 \le \text{nums.length} \le 5000$
- $-10^4 \le \text{nums[i]} \le 10^4$
- $-10^4 < \mathsf{target} < 10^4$

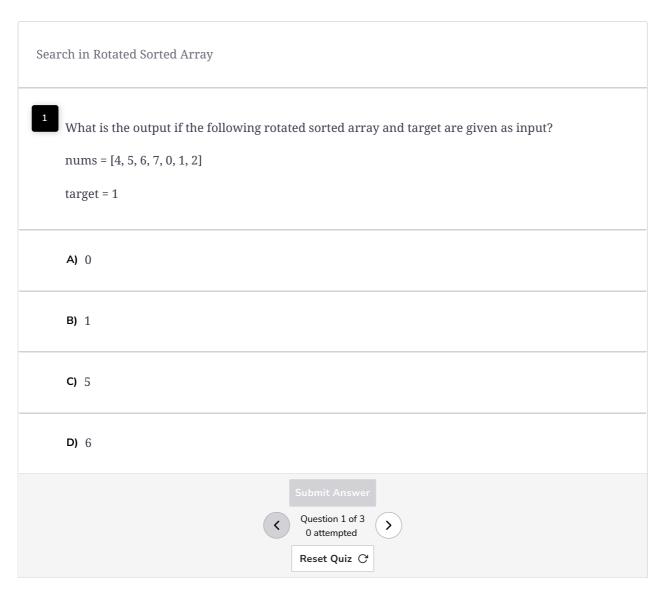
## **Examples**





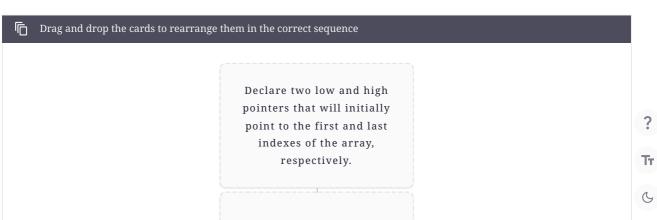
## Understand the problem

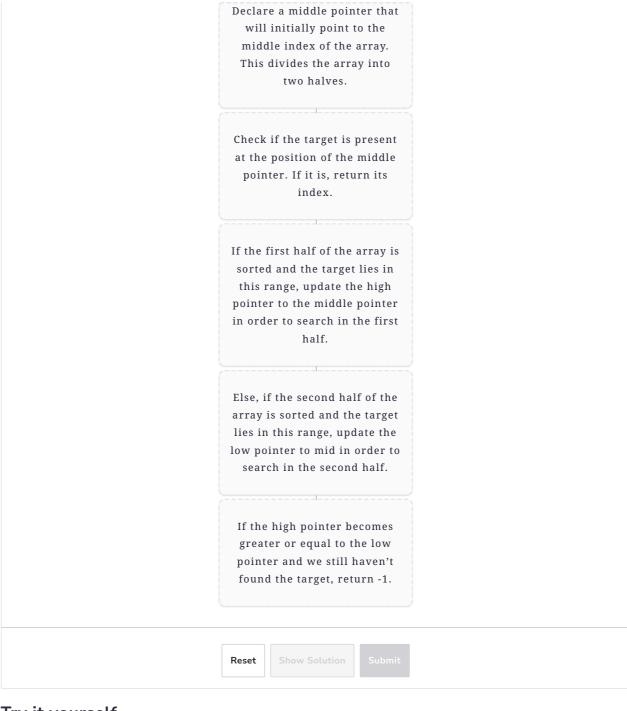
Let's take a moment to make sure you've correctly understood the problem. The quiz below helps us to check if you're solving the correct problem:



## Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.





# Try it yourself

Implement your solution in the following coding playground:



```
usercode > main.java
1 import java.util.*;
2 public class Main{
3
      public static int binarySearchRotated(List<Integer> nums, int target) {
5
         // Write your code here
7
         return -1;
8
9 }
                                                                                                               ?
                                                                                                              Tr
                                                                                                               6
                                                                           Powered by Al
                                                                                                   Submit
```

