```
#include <stdio.h>
#include <stdlib.h>

int binarySearch(int arr[], int start, int end, int majorityElement)
{
	if(end >= start)
	{
		int middle = (start + end) / 2;
		if ( middle == 0 || majorityElement > arr[middle - 1] && arr[middle] ==
majorityElement)
			return middle;
		else if (majorityElement > arr[middle])
			return binarySearch(arr, (middle + 1), end, majorityElement);
		return binarySearch(arr, start, (middle - 1), majorityElement);
	}
}

int checkMajority(int arr[], int size, int majorityElement)
{
	int index = binarySearch(arr, 0, size-1, majorityElement);
	if (index == -1)
		return 0;
	return (((index + size/2) <= (size -1)) && arr[index + size/2] == majorityElement)?
	1: 0;
}

int main()
{
	int *arr, size, ele;
	printf("Enter size of the array\n");
	scanf("%d", &size);
	printf("Enter elements in array\n");
	for(int index = 0; index < size; index++)
		scanf("%d", &arr[index]);
	printf("Enter element to search\n");
	scanf("%d", &ele);
	checkMajority(arr, size, ele)?
	printf("%d appears more than n/2 times", ele) :
	printf("%d doesn't appears more than n/2 times", ele);
	return 0;
}
```
Time complexity: O(logn)
space complexity: O(logn)