

DEVADHARSHINI G 2024-CSE**D2****Started on** Wednesday, 17 September 2025, 10:53 AM**State** Finished**Completed on** Wednesday, 8 October 2025, 10:31 AM**Time taken** 20 days 23 hours**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

Question 1 | Correct Mark 1.00 out of 1.00**Problem Statement**

Given an array of 1s and 0s this has all 1s first followed by all 0s. Aim is to find the number of 0s. Write a program using Divide and Conquer to Count the number of zeroes in the given array.

Input Format

First Line Contains Integer m – Size of array

Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 // Function to find the index of the first 0 using binary search
4 int findFirstZero(int arr[], int low, int high) {
5     if (high >= low) {
6         int mid = low + (high - low) / 2;
7
8         // Check if mid is the first 0
9         if ((mid == 0 || arr[mid - 1] == 1) && arr[mid] == 0)
10            return mid;
11
12        // If the element is 1, search in the right half
13        if (arr[mid] == 1)
14            return findFirstZero(arr, mid + 1, high);
15
16        // Else, search in the left half
17        return findFirstZero(arr, low, mid - 1);
18    }
19
20    // No zero found
21    return -1;
22}
23
24 int main() {
25     int m;
26     scanf("%d", &m);
27     int arr[m];
28
29     for (int i = 0; i < m; i++) {
30         scanf("%d", &arr[i]);
31     }
32
33     int index = findFirstZero(arr, 0, m - 1);
34     int count = (index == -1) ? 0 : (m - index);
35
36     printf("%d\n", count);
37     return 0;
38 }
39
```

	Input	Expected	Got	
✓	5 1 1 1 0 0	2	2	✓
✓	10 1 1 1 1 1 1 1 1 1	0	0	✓
✓	8 0 0 0 0 0 0 0 0	8	8	✓
✓	17 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

DEVADHARSHINI G 2024-CSE**D2****Started on** Wednesday, 17 September 2025, 11:22 AM**State** Finished**Completed on** Wednesday, 8 October 2025, 10:32 AM**Time taken** 20 days 23 hours**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

Question 1 | Correct Mark 1.00 out of 1.00

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

Input: `nums = [3,2,3]`

Output: 3

Example 2:

Input: `nums = [2,2,1,1,1,2,2]`

Output: 2

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 104`
- `-231 <= nums[i] <= 231 - 1`

For example:

Input	Result
3	3
3 2 3	
7	2
2 2 1 1 1 2 2	

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int majorityElement(int* nums, int numsSize) {
4     int count = 0;
5     int candidate = 0;
6
7     for (int i = 0; i < numsSize; i++) {
8         if (count == 0) {
9             candidate = nums[i];
10            count = 1;
11        } else if (nums[i] == candidate) {
12            count++;
13        } else {
14            count--;
15        }
16    }
17    return candidate;
18}
19
20 int main() {
21     int n;
22     scanf("%d", &n);
23     int nums[n];
24
25     for (int i = 0; i < n; i++) {
26         scanf("%d", &nums[i]);
27     }
28 }
```

```
27 }  
28  
29     int result = majorityElement(nums, n);  
30     printf("%d\\n", result);  
31  
32     return 0;  
33 }  
34 }
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

DEVADHARSHINI G 2024-CSE**D2****Started on** Wednesday, 8 October 2025, 10:33 AM**State** Finished**Completed on** Wednesday, 8 October 2025, 10:49 AM**Time taken** 15 mins 28 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

Question 1 | Correct Mark 1.00 out of 1.00**Problem Statement:**

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Value for x

Output Format

First Line Contains Integer – Floor value for x

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int n, x;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &arr[i]);
10    }
11
12    scanf("%d", &x);
13
14    int low = 0, high = n - 1;
15    int floorValue = -1;
16
17    while (low <= high) {
18        int mid = (low + high) / 2;
19
20        if (arr[mid] == x) {
21            floorValue = arr[mid];
22            break;
23        } else if (arr[mid] < x) {
24            floorValue = arr[mid];
25            low = mid + 1;
26        } else {
27            high = mid - 1;
28        }
29    }
30
31    printf("%d\n", floorValue);
32
33    return 0;
34}
35
```

	Input	Expected	Got	
✓	6 1 2 8 10 12 19 5	2	2	✓
✓	5 10 22 85 108 129 100	85	85	✓
✓	7 3 5 7 9 11 13 15 10	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

DEVADHARSHINI G 2024-CSE**D2****Started on** Wednesday, 8 October 2025, 10:52 AM**State** Finished**Completed on** Wednesday, 8 October 2025, 10:56 AM**Time taken** 3 mins 23 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

Question 1 | Correct Mark 1.00 out of 1.00**Problem Statement:**

Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Sum Value

Output Format

First Line Contains Integer – Element1

Second Line Contains Integer – Element2 (Element 1 and Elements 2 together sums to value "x")

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int main() {
4     int n, x;
5     scanf("%d", &n);
6     int arr[n];
7     for (int i = 0; i < n; i++)
8         scanf("%d", &arr[i]);
9     scanf("%d", &x);
10
11    int left = 0, right = n - 1;
12    int found = 0;
13    int element1 = -1, element2 = -1;
14
15    while (left < right && !found) {
16        int sum = arr[left] + arr[right];
17        if (sum == x) {
18            element1 = arr[left];
19            element2 = arr[right];
20            found = 1;
21        } else if (sum < x) {
22            left++;
23        } else {
24            right--;
25        }
26    }
27
28    if (found) {
29        printf("%d\n%d\n", element1, element2);
30    } else {
31        printf("No\n");
32    }
33
34    return 0;
35}
36

```

	Input	Expected	Got	
✓	4 2 4 8 10 14	4 10	4 10	✓
✓	5 2 4 6 8 10 100	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

DEVADHARSHINI G 2024-CSE**D2****Started on** Wednesday, 8 October 2025, 10:56 AM**State** Finished**Completed on** Wednesday, 8 October 2025, 11:12 AM**Time taken** 15 mins 52 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

Question 1 | Correct Mark 1.00 out of 1.00

Write a Program to Implement the Quick Sort Algorithm

Input Format:

The first line contains the no of elements in the list-n

The next n lines contain the elements.

Output:

Sorted list of elements

For example:

Input	Result
5	12 34 67 78 98
67 34 12 98 78	

Answer:

```

1 #include <stdio.h>
2
3 void swap(int *a, int *b) {
4     int t = *a;
5     *a = *b;
6     *b = t;
7 }
8
9 int partition(int arr[], int low, int high) {
10    int pivot = arr[high];
11    int i = low - 1;
12    for (int j = low; j < high; j++) {
13        if (arr[j] <= pivot) {
14            i++;
15            swap(&arr[i], &arr[j]);
16        }
17    }
18    swap(&arr[i + 1], &arr[high]);
19    return i + 1;
20 }
21
22 void quickSort(int arr[], int low, int high) {
23    if (low < high) {
24        int pi = partition(arr, low, high);
25        quickSort(arr, low, pi - 1);
26        quickSort(arr, pi + 1, high);
27    }
28 }
29
30 int main() {
31    int n;
32    scanf("%d", &n);
33    int arr[n];
34    for (int i = 0; i < n; i++)
35        scanf("%d", &arr[i]);
36
37    quickSort(arr, 0, n - 1);
38
39    for (int i = 0; i < n; i++) {
40        printf("%d", arr[i]);
41        if (i != n - 1)
42            printf(" ");
43    }
44    printf("\n");
45
46    return 0;

```

47 }
48

	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 90 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

/