



SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 17 September 2025, 8:08 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:16 AM
Time taken	7 mins 22 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 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  int countZeroes(int arr[], int left, int right) {
4      if (left > right) {
5          return 0;
6      }
7      if (arr[left] == 0 && arr[right] == 0) {
8          return right - left + 1;
9      }
10
11     int mid = (left + right) / 2;
12     if (arr[mid] == 0) {
13         return (right - mid + 1) + countZeroes(arr, left, mid - 1);
14     }
15     return countZeroes(arr, mid + 1, right);
16 }
17
18 int main() {
19     int m, i;
20     scanf("%d", &m);
21
22     int arr[m];
23     for (i = 0; i < m; i++) {
24         scanf("%d", &arr[i]);
25     }
26
27     int result = countZeroes(arr, 0, m - 1);
28     printf("%d\n", result);
29
30     return 0;
31 }
32
33

```

	Input	Expected	Got	
✓	5	2	2	✓
	1			
	1			
	1			
	0			
	0			

	Input	Expected	Got	
✓	10 1 1 1 1 1 1 1 1 1 1 1 1	0	0	✓
✓	8 0 0 0 0 0 0 0 0 0 0	8	8	✓
✓	17 1 1 1 1 1 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](#)



SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 17 September 2025, 8:19 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:25 AM
Time taken	5 mins 44 secs
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 2 3	3
7 2 2 1 1 1 2 2	2

**Answer:** (penalty regime: 0 %)

```

1  #include <stdio.h>
2
3  int majorityElement(int* nums, int numsSize) {
4      int candidate = nums[0], count = 1;
5
6      for (int i = 1; i < numsSize; i++) {
7          if (nums[i] == candidate) {
8              count++;
9          } else {
10             count--;
11             if (count == 0) {
12                 candidate = nums[i];
13                 count = 1;
14             }
15         }
16     }
17
18     return candidate;
19 }
20
21 int main() {
22     int n;
23     scanf("%d", &n);
24
25     int nums[n];
26     for (int i = 0; i < n; i++) {
27         scanf("%d", &nums[i]);
28     }
29
30     // Call the majorityElement function and print the result
31     printf("%d\n", majorityElement(nums, n));
32
33     return 0;
34 }
```

35 |

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)





SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 17 September 2025, 8:25 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:26 AM
Time taken	46 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 findFloor(int* nums, int n, int x) {
4      int low = 0, high = n - 1;
5      int result = -1;
6
7      while (low <= high) {
8          int mid = low + (high - low) / 2;
9
10         if (nums[mid] == x) {
11             return nums[mid];
12         } else if (nums[mid] < x) {
13             result = nums[mid];
14             low = mid + 1;
15         } else {
16             high = mid - 1;
17         }
18     }
19
20     return result;
21 }
22
23 int main() {
24     int n, x;
25     scanf("%d", &n);
26
27     int nums[n];
28     for (int i = 0; i < n; i++) {
29         scanf("%d", &nums[i]);
30     }
31
32     scanf("%d", &x);
33
34     printf("%d\n", findFloor(nums, n, x));
35
36     return 0;
37 }
38

```

	Input	Expected	Got	
✓	6	2	2	✓
	1			
	2			
	8			
	10			
	12			
	19			
	5			



	Input	Expected	Got	
✓	5	85	85	✓
	10			
	22			
	85			
	108			
	129			
	100			
✓	7	9	9	✓
	3			
	5			
	7			
	9			
	11			
	13			
	15			
	10			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 17 September 2025, 8:26 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:27 AM
Time taken	1 min 7 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 findPairRecursive(int* arr, int low, int high, int x) {
4      if (low >= high) {
5          return 0;
6      }
7
8      int sum = arr[low] + arr[high];
9
10     if (sum == x) {
11         printf("%d\n", arr[low]);
12         printf("%d\n", arr[high]);
13         return 1;
14     }
15
16     if (sum < x) {
17         return findPairRecursive(arr, low + 1, high, x);
18     } else {
19         return findPairRecursive(arr, low, high - 1, x);
20     }
21 }
22
23 int main() {
24     int n, x;
25     scanf("%d", &n);
26
27     int arr[n];
28     for (int i = 0; i < n; i++) {
29         scanf("%d", &arr[i]);
30     }
31
32     scanf("%d", &x);
33
34     if (!findPairRecursive(arr, 0, n - 1, x)) {
35         printf("No\n");
36     }
37
38     return 0;
39 }
40

```

	Input	Expected	Got	
✓	4	4	4	✓
	2	10	10	
	4			
	8			
	10			
	14			

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 17 September 2025, 8:27 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:28 AM
Time taken	50 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 temp = *a;
5      *a = *b;
6      *b = temp;
7  }
8
9  int partition(int arr[], int low, int high) {
10     int pivot = arr[high];
11     int i = low - 1;
12
13     for (int j = low; j < high; j++) {
14         if (arr[j] <= pivot) {
15             i++;
16             swap(&arr[i], &arr[j]);
17         }
18     }
19     swap(&arr[i + 1], &arr[high]);
20     return i + 1;
21 }
22
23 void quickSort(int arr[], int low, int high) {
24     if (low < high) {
25         int pi = partition(arr, low, high);
26         quickSort(arr, low, pi - 1);
27         quickSort(arr, pi + 1, high);
28     }
29 }
30
31 int main() {
32     int n;
33     scanf("%d", &n);
34
35     int arr[n];
36     for (int i = 0; i < n; i++) {
37         scanf("%d", &arr[i]);
38     }
39
40     quickSort(arr, 0, n - 1);
41
42     for (int i = 0; i < n; i++) {
43         printf("%d ", arr[i]);
44     }
45
46     return 0;
47 }
48
49

```

	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 11 10 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](#)