



Started on	Tuesday, 23 September 2025, 1:44 PM
State	Finished
Completed on	Monday, 17 November 2025, 8:33 PM
Time taken	55 days 6 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 int findFirstZero(int arr[], int m) {
3     int left = 0, right = m - 1;
4     int firstZeroIndex = -1;
5     while (left <= right) {
6         int mid = left + (right - left) / 2;
7         if (arr[mid] == 0) {
8             firstZeroIndex = mid;
9             right = mid - 1;
10        } else {
11            left = mid + 1;
12        }
13    }
14    return firstZeroIndex;
15 }
16 int main() {
17     int m;
18     scanf("%d", &m);
19     int arr[m];
20     for (int i = 0; i < m; i++) {
21         scanf("%d", &arr[i]);
22     }
23     int firstZero = findFirstZero(arr, m);
24     if (firstZero == -1) {
25         printf("0\n");
26     } else {
27         printf("%d\n", m - firstZero);
28     }
29     return 0;
30 }
31
32

```

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

	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](#)



SWETHA VEERAMANI 2024-AIML ▾

S2

Started on	Tuesday, 23 September 2025, 1:52 PM
State	Finished
Completed on	Monday, 17 November 2025, 8:34 PM
Time taken	55 days 6 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 2 3	3
7 2 2 1 1 1 2 2	2

Answer: (penalty regime: 0 %)

```

1  #include <stdio.h>
2  int majorityElement(int nums[], int n) {
3      int count = 0;
4      int candidate = 0;
5
6      for (int i = 0; i < n; i++) {
7          if (count == 0) {
8              candidate = nums[i];
9              count = 1;
10         } else if (nums[i] == candidate) {
11             count++;
12         } else {
13             count--;
14         }
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
29     int result = majorityElement(nums, n);
30
31     printf("%d\n", result);
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](#)



SWETHA VEERAMANI 2024-AIML ▾

S2

Started on	Tuesday, 23 September 2025, 2:00 PM
State	Finished
Completed on	Monday, 17 November 2025, 8:34 PM
Time taken	55 days 6 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 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 arr[], int n, int x) {
4      int left = 0, right = n - 1;
5      int floor = -1; // default if no floor found
6
7      while (left <= right) {
8          int mid = left + (right - left) / 2;
9
10         if (arr[mid] == x) {
11             return arr[mid]; // floor is x itself
12         }
13         else if (arr[mid] < x) {
14             floor = arr[mid]; // possible floor
15             left = mid + 1; // try to find bigger floor on right
16         }
17         else {
18             right = mid - 1; // go left for smaller values
19         }
20     }
21
22     return floor;
23 }
24
25 int main() {
26     int n, x;
27     scanf("%d", &n);
28
29     int arr[n];
30     for (int i = 0; i < n; i++)
31         scanf("%d", &arr[i]);
32
33     scanf("%d", &x);
34
35     printf("%d\n", findFloor(arr, n, x));
36
37     return 0;
38 }
39
```


	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](#)



Started on	Tuesday, 23 September 2025, 2:07 PM
State	Finished
Completed on	Monday, 17 November 2025, 8:35 PM
Time taken	55 days 6 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 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  void findPairRecursive(int arr[], int left, int right, int x) {
4      if (left >= right) {
5          printf("No\n");
6          return;
7      }
8
9      int sum = arr[left] + arr[right];
10
11     if (sum == x) {
12         printf("%d\n%d\n", arr[left], arr[right]);
13         return;
14     }
15     else if (sum < x) {
16         findPairRecursive(arr, left + 1, right, x);
17     }
18     else {
19         findPairRecursive(arr, left, right - 1, x);
20     }
21 }
22
23 int main() {
24     int n;
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     int x;
33     scanf("%d", &x);
34
35     findPairRecursive(arr, 0, n - 1, x);
36
37     return 0;
38 }
39

```

	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](#)



Started on	Tuesday, 23 September 2025, 2:15 PM
State	Finished
Completed on	Monday, 17 November 2025, 8:36 PM
Time taken	55 days 6 hours
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; *a = *b; *b = t;
5  }
6
7  int partition(int arr[], int low, int high) {
8      int pivot = arr[high], i = low - 1;
9      for (int j = low; j < high; j++) {
10         if (arr[j] < pivot) swap(&arr[++i], &arr[j]);
11     }
12     swap(&arr[i + 1], &arr[high]);
13     return i + 1;
14 }
15
16 void quickSort(int arr[], int low, int high) {
17     if (low < high) {
18         int pi = partition(arr, low, high);
19         quickSort(arr, low, pi - 1);
20         quickSort(arr, pi + 1, high);
21     }
22 }
23
24 int main() {
25     int n;
26     scanf("%d", &n);
27     int arr[n];
28     for (int i = 0; i < n; i++) scanf("%d", &arr[i]);
29
30     quickSort(arr, 0, n - 1);
31
32     for (int i = 0; i < n; i++) printf("%d ", arr[i]);
33     printf("\n");
34     return 0;
35 }
36

```

	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	✓

	Input	Expected	Got	
✓	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](#)