



ALVIN B 2024-CSE ▾

A2

**Started on** Monday, 29 September 2025, 1:43 PM**State** Finished**Completed on** Monday, 29 September 2025, 2:02 PM**Time taken** 19 mins 13 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 findFirstZero(int arr[], int low, int high) {
4     if (high >= low) {
5         int mid = low + (high - low) / 2;
6
7         if ((mid == 0 || arr[mid - 1] == 1) && arr[mid] == 0) {
8             return mid;
9         }
10
11        if (arr[mid] == 1) {
12            return findFirstZero(arr, mid + 1, high);
13        }
14        else {
15            return findFirstZero(arr, low, mid - 1);
16        }
17    }
18    return -1;
19 }
20 int countZeroes(int arr[], int n) {
21     int firstZeroIndex = findFirstZero(arr, 0, n - 1);
22
23     if (firstZeroIndex == -1) {
24         return 0;
25     }
26     return (n - firstZeroIndex);
27 }
28
29 int main() {
30     int m;
31     scanf("%d", &m);
32
33     int arr[m];
34     for (int i = 0; i < m; i++) {
35         scanf("%d", &arr[i]);
36     }
37
38     printf("%d\n", countZeroes(arr, m));
39
40     return 0;
41 }
```

	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	0	0	✓
✓	8 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 0 0	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

**Started on** Monday, 29 September 2025, 2:02 PM

**State** Finished

**Completed on** Monday, 29 September 2025, 2:09 PM

**Time taken** 6 mins 53 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 * 10^4`
- `-2^31 <= nums[i] <= 2^31 - 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 #include <stdlib.h>
3 int majorityElement(int nums[], int n) {
4     int candidate = 0;
5     int count = 0;
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, i;
22
23     if (scanf("%d", &n) != 1 || n <= 0) {
24         printf("Invalid input for array size.\n");
25         return 1;
26     }
27     int* nums = (int*)malloc(n * sizeof(int));
28     if (nums == NULL) {
29         printf("Memory allocation failed.\n");
30         return 1;
31     }
32
33     for (i = 0; i < n; i++) {
34         scanf("%d", &nums[i]);
35     }
36 }
```

```
35 }  
36     int result = majorityElement(nums, n);  
37     printf("%d\\n", result);  
38     free(nums);  
39  
40     return 0;  
41 }
```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



ALVIN B 2024-CSE ▾

A2

**Started on** Monday, 29 September 2025, 2:10 PM

**State** Finished

**Completed on** Monday, 29 September 2025, 2:13 PM

**Time taken** 2 mins 55 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 #include <stdlib.h>
3
4 int findFloor(int arr[], int n, int x) {
5     int low = 0;
6     int high = n - 1;
7     int floor_val = -1;
8     while (low <= high) {
9         int mid = low + (high - low) / 2;
10    if (arr[mid] == x) {
11        return arr[mid];
12    } else if (arr[mid] < x) {
13        floor_val = arr[mid];
14        low = mid + 1;
15    } else {
16        high = mid - 1;
17    }
18 }
19 return floor_val;
20 }
21
22 int main() {
23     int n, i, x;
24     scanf("%d", &n);
25     int* arr = (int*)malloc(n * sizeof(int));
26     for (i = 0; i < n; i++) {
27         scanf("%d", &arr[i]);
28     }
29     scanf("%d", &x);
30     int floor_value = findFloor(arr, n, x);
31     printf("%d\n", floor_value);
32     free(arr);
33     return 0;
34 }
35

```

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

	Input	Expected	Got	
✓	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](#)



ALVIN B 2024-CSE

A2

**Started on** Monday, 29 September 2025, 2:13 PM**State** Finished**Completed on** Monday, 29 September 2025, 2:22 PM**Time taken** 8 mins 29 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 #include <stdlib.h>
3
4 int find_pair_recursive(int arr[], int low, int high, int x) {
5     if (low >= high) {
6         return 0;
7     }
8
9     int sum = arr[low] + arr[high];
10
11    if (sum == x) {
12        printf("%d\n", arr[low]);
13        printf("%d\n", arr[high]);
14        return 1;
15    } else if (sum < x) {
16        return find_pair_recursive(arr, low + 1, high, x);
17    } else {
18        return find_pair_recursive(arr, low, high - 1, x);
19    }
20}
21
22 int main() {
23     int n, i, x;
24     scanf("%d", &n);
25     int* arr = (int*)malloc(n * sizeof(int));
26     if (arr == NULL) {
27         return 1;
28     }
29     for (i = 0; i < n; i++) {
30         scanf("%d", &arr[i]);
31     }
32     scanf("%d", &x);
33
34     if (!find_pair_recursive(arr, 0, n - 1, x)) {
35         printf("No\n");
36     }
37
38     free(arr);
39     return 0;
40 }
41
```

	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](#)



7 ALVIN B 2024-CSE

A2

**Started on** Monday, 29 September 2025, 2:22 PM**State** Finished**Completed on** Monday, 29 September 2025, 2:37 PM**Time taken** 15 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

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 #include <stdlib.h>
3
4 void swap(int* a, int* b) {
5     int temp = *a;
6     *a = *b;
7     *b = temp;
8 }
9
10 int partition(int arr[], int low, int high) {
11     int pivot = arr[high];
12     int i = (low - 1);
13     int j;
14
15     for (j = low; j < high; j++) {
16         if (arr[j] <= pivot) {
17             i++;
18             swap(&arr[i], &arr[j]);
19         }
20     }
21     swap(&arr[i + 1], &arr[high]);
22     return (i + 1);
23 }
24
25 void quickSort(int arr[], int low, int high) {
26     if (low < high) {
27         int pi = partition(arr, low, high);
28         quickSort(arr, low, pi - 1);
29         quickSort(arr, pi + 1, high);
30     }
31 }
32
33 void printArray(int arr[], int size) {
34     int i;
35     for (i = 0; i < size; i++) {
36         printf("%d ", arr[i]);
37     }
38     printf("\n");
39 }
40
41 int main() {
42     int n, i;
43     scanf("%d", &n);
44     int* arr = (int*)malloc(n * sizeof(int));
45     if (arr == NULL) {
46         return 1;
47     }
48     for (i = 0; i < n; i++) {
49         scanf("%d", &arr[i]);
50     }
51     quickSort(arr, 0, n - 1);

```

	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](#)