



KAAJAL G 2024-CSE ▾

K2**Started on** Wednesday, 17 September 2025, 3:39 PM**State** Finished**Completed on** Wednesday, 17 September 2025, 4:37 PM**Time taken** 57 mins 4 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 par(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
20     swap(&arr[i + 1], &arr[high]);
21     return i + 1;
22 }
23
24 void qt(int arr[], int low, int high) {
25     if (low < high) {
26         int pi = par(arr, low, high);
27
28         qt(arr, low, pi - 1);
29         qt(arr, pi+1, high);
30     }
31 }
32
33 int main() {
34     int n;
35     scanf("%d", &n);
36
37     int arr[n];
38     for (int i = 0; i < n; i++)
39         scanf("%d", &arr[i]);
40
41     qt(arr, 0, n - 1);
42
43     for (int i = 0; i < n; i++)
44         printf("%d ", arr[i]);
45
46     printf("\n");
47     return 0;
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](#)



KAAJAL G 2024-CSE ▾

K2

Started on Wednesday, 17 September 2025, 3:39 PM

State Finished

Completed on Wednesday, 17 September 2025, 3:39 PM

Time taken 25 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 findPair(int arr[], int left, int right, int x, int* a, int* b) {
4      if (left >= right)
5          return 0;
6
7      int sum = arr[left] + arr[right];
8
9      if (sum == x) {
10         *a = arr[left];
11         *b = arr[right];
12         return 1;
13     } else if (sum < x) {
14         return findPair(arr, left + 1, right, x, a, b);
15     } else {
16         return findPair(arr, left, right - 1, x, a, b);
17     }
18 }
19
20 int main() {
21     int n, x;
22     scanf("%d", &n);
23
24     int arr[n];
25     for (int i = 0; i < n; i++)
26         scanf("%d", &arr[i]);
27
28     scanf("%d", &x);
29
30     int a, b;
31     if (findPair(arr, 0, n - 1, x, &a, &b)) {
32         printf("%d\n%d\n", a, b);
33     } else {
34         printf("No\n");
35     }
36
37     return 0;
38 }

```

	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](#)



KAAJAL G 2024-CSE ▾

K2

Started on Wednesday, 17 September 2025, 3:38 PM

State Finished

Completed on Wednesday, 17 September 2025, 3:38 PM

Time taken 21 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 arr[], int low, int high, int x) {
4      if (low > high)
5          return -1; // No floor exists
6
7      int mid = low + (high - low) / 2;
8
9      if (arr[mid] == x)
10         return arr[mid];
11
12     if (arr[mid] > x)
13         return findFloor(arr, low, mid - 1, x);
14
15     int temp = findFloor(arr, mid + 1, high, x);
16     return (temp != -1) ? temp : arr[mid];
17 }
18
19 int main() {
20     int n, x;
21     scanf("%d", &n);
22
23     int arr[n];
24     for (int i = 0; i < n; i++)
25         scanf("%d", &arr[i]);
26
27     scanf("%d", &x);
28
29     int floorValue = findFloor(arr, 0, n - 1, x);
30     printf("%d\n", floorValue);
31
32     return 0;
33 }
34

```

	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](#)



KAAJAL G 2024-CSE ▾

K2

Started on Wednesday, 17 September 2025, 3:38 PM

State Finished

Completed on Wednesday, 17 September 2025, 3:38 PM

Time taken 12 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 count = 0, candidate = 0;
5 |
6 |     for (int i = 0; i < numsSize; i++) {
7 |         if (count == 0)
8 |             candidate = nums[i];
9 |         count += (nums[i] == candidate) ? 1 : -1;
10 |    }
11 |
12 |    return candidate;
13 | }
14 |
15 | int main() {
16 |     int n;
17 |     scanf("%d", &n);
18 |
19 |     int nums[n];
20 |     for (int i = 0; i < n; i++)
21 |         scanf("%d", &nums[i]);
22 |
23 |     int result = majorityElement(nums, n);
24 |     printf("%d\n", result);
25 |
26 |     return 0;
27 | }
28 |

```

	Input	Expected	Got	
✓	3 3 2 3	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



Started on	Wednesday, 17 September 2025, 3:31 PM
State	Finished
Completed on	Wednesday, 17 September 2025, 3:37 PM
Time taken	6 mins 3 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         if (arr[mid] == 1)
11             return findFirstZero(arr, mid + 1, high);
12         else
13             return findFirstZero(arr, low, mid - 1);
14     }
15     return -1;
16 }
17
18 int main() {
19     int m;
20     scanf("%d", &m);
21
22     int arr[m];
23     for (int i = 0; i < m; i++)
24         scanf("%d", &arr[i]);
25
26     int firstZeroIndex = findFirstZero(arr, 0, m - 1);
27     int zeroCount = (firstZeroIndex == -1) ? 0 : m - firstZeroIndex;
28
29     printf("%d\n", zeroCount);
30     return 0;
31 }

```

	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](#)