



<

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



## 2-Majority Element



Started on Wednesday, 17 September 2025, 8:45 AM

State Finished

Completed on Wednesday, 17 September 2025, 9:12 AM

Time taken 27 mins 30 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 Correct Mark 1.00 out of 1.00 [Flag question](#)

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:

- $n == \text{nums.length}$
- $1 \leq n \leq 5 \cdot 10^4$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$

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 int main(){
3     int n,r;
4     scanf("%d",&n);
5     int m=n/2;
6     int arr[n];
7     for(int i=0;i<n;i++){
8         scanf("%d",&arr[i]);
9     }
10    int count=0;
11    for(int i=0;i<n;i++){
12        for(int j=0;j<n;j++){
13            if(arr[i]==arr[j]){
14                count++;
15            }
16            if(count>m){
17                r=arr[j];
18            }
19        }
20    }
21    printf("%d",r);
22 }
23 }
```

	Input	Expected	Got	
✓	3	3	3	✓
	3 2 3			

Passed all tests! ✓

Correct

### 3-Finding Floor Value

Started on Wednesday, 17 September 2025, 8:38 AM

State: Finished

Completed on Wednesday, 17 September 2025, 8:55 AM

Time taken: 17 mins 39 secs

Marks: 1.00/1.00

Grade: 10.00 out of 10.00 (100%)

Question 1 : Correct Mark 1.00 out of 1.00 [Flag question](#)

#### 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 int main()
3 {
4     int arr[100],n,x;
5     scanf("%d",&n);
6     for(int i=0;i<n;i++)
7     {
8         scanf("%d",&arr[i]);
9     }
10    scanf("%d",&x);
11    for(int i=0;i<n;i++)
12    {
13        if(x<=arr[i])
14        {
15            printf("%d",arr[i-1]);
16            break;
17        }
18    }
19 }
20 }
```

	Input	Expected	Got	
✓	6 1 2 8 10 12 18 5	2	2	✓
✓	5 10 22 85 108 129 160	85	85	✓
✓	7 3 5 7 9 11 13 15 16	9	9	✓

Passed all tests! ✓

Correct



## 4-Two Elements sum to x

Started on Wednesday, 17 September 2025, 8:20 AM

State Finished

Completed on Wednesday, 17 September 2025, 8:38 AM

Time taken 18 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 [Flag question](#)

### 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 int main(){
3     int n;
4     scanf("%d",&n);
5     int arr[n];
6     for(int i=0;i<n;i++){
7         scanf("%d",&arr[i]);
8     }
9     int x;
10    scanf("%d",&x);
11    int res=arr[1]-arr[n-1];
12    if(x==res){
13        printf("%d\n",arr[1]);
14        printf("%d",arr[n-1]);
15    }
16    }
17    else{
18        printf("No");
19    }
20 }
21
22
23 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



# 5-Implementation of Quick Sort

Started on Wednesday, 17 September 2025, 9:13 AM

State Finished

Completed on Wednesday, 17 September 2025, 9:32 AM

Time taken 19 mins 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 [Flag question](#)

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 int main(){
4     int n;
5     scanf("%d",&n);
6     int arr[n];
7     for(int i=0;i<n;i++){
8         scanf("%d",&arr[i]);
9     }
10
11     int cmp(const void *a,const void *b){
12         return(*(int *)a)-*(int *)b;
13     }
14     qsort(arr,n,sizeof(int),cmp);
15     for(int i=0;i<n;i++){
16         printf("%d ",arr[i]);
17     }
18 }
```

Input	Expected	Got	
✓ 5 12 34 67 78 98	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