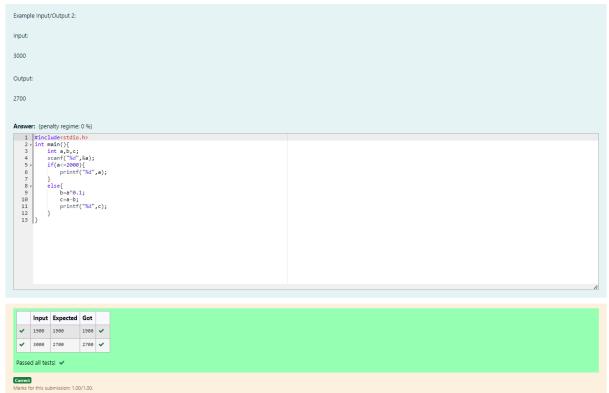
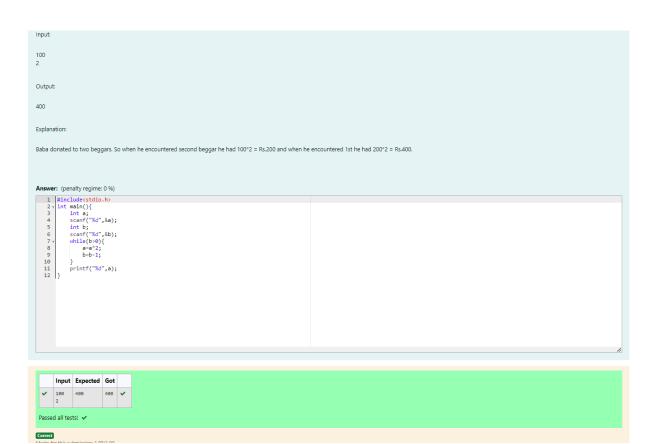
WEEK_1





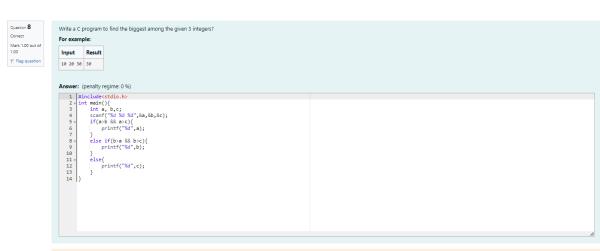
3.

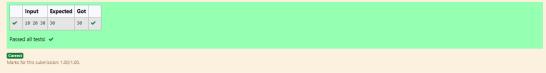




5.

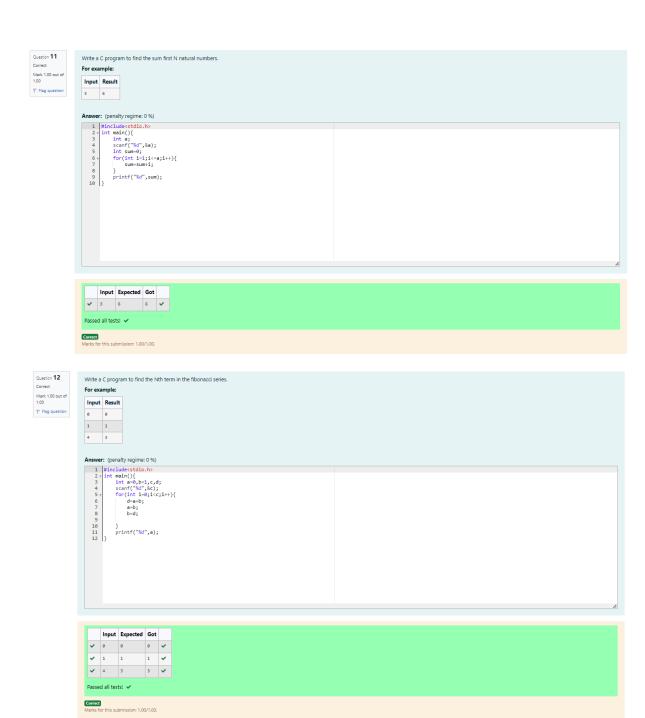


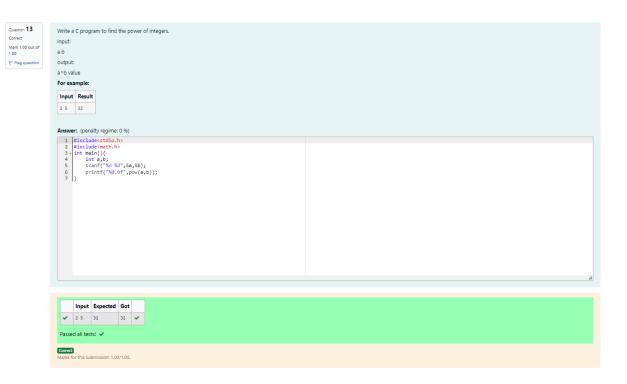




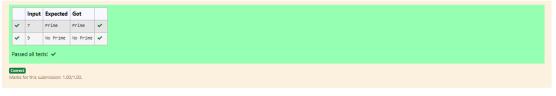






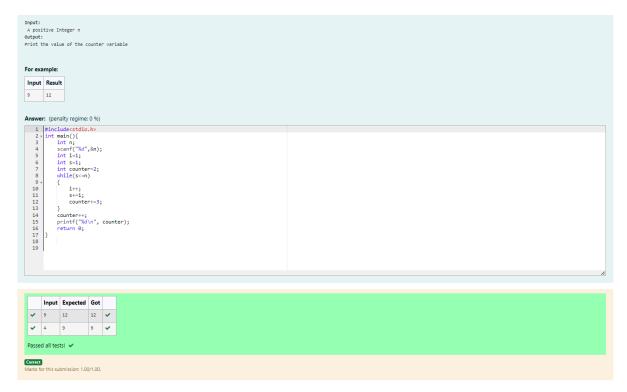


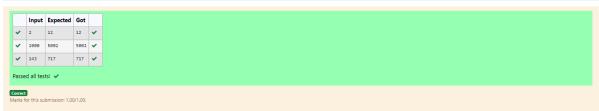
Question 14
Correct
Mark 1.00 out of 1.00
P Flag question



TIME COMPLEXITY

1.





```
Convert the following algorithm into a program and find its time

Convert the following algorithm into a program and find its time

Complexity using counter method.

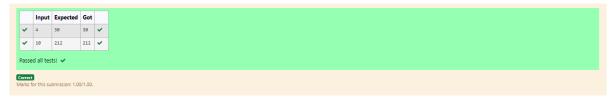
What to Constant

The growthout

This growthout

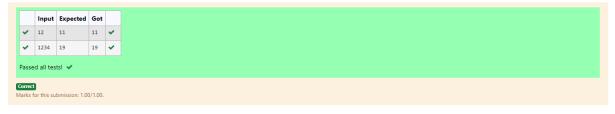
This growthout

The growthout
```



Finish review

```
Question 1
Correct
Mark 1.00 out of 1.00
F Flag question
```

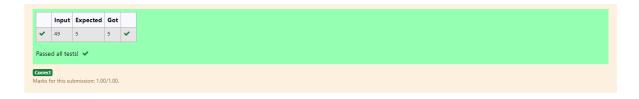


GREEDY METHOD

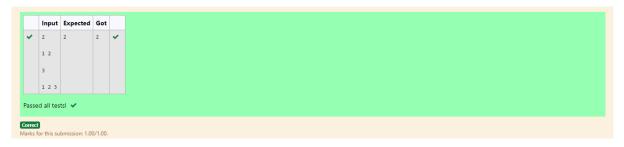
```
Output

Answer: (penally regime: 0 %)

Answer: (penally regime
```



```
Question 1
                        Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.
                        Each child i has a greed factor g[i], which is the minimum size of a cookie that the child will be content with: and each cookie j has a size s[j]. If s[j] >= g[i], we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.
 Mark 1.00 out of 1.00
₹ Flag question
                        Example 1:
                        Input:
                        123
                        2
                        11
                        Output:
                        Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3.
                        And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.
                        You need to output 1.
                        Constraints:
                        1 <= g.length <= 3 * 10^4
                        0 <= s.length <= 3 * 10^4
                        1 <= g[i], s[j] <= 2^31 - 1
```



```
A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a distance to burn out his calories.

If he has eaten i burgers with c calories each, then he has to run at least 3! * c kilometers to burn out the calories. For example, if he ate 3 burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are (3* 1) + (3* * 2) + (3* * 2) + 1 + 9 + 18 - 28. But this is not the siniahum, so need to try out other orders of consumption and choose the minimum value. Determine the minimum distance he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm.Apply greedy approach to solve the problem.

Input Fornat

Print: Minimum number of burgers

Sample Input

For example:

Test Input Result

Test Case 1 3 18

1 3 2

Answer: (penalty reqime: 0 %6)
```

Test Case 1 3 18 18 18 Test Case 2 4 389 38 7 4 9 6
7 4 9 6
Test Case 3 3 76 76 76

```
Given an array of N integer, we have to maximize the sum of arr[i] * i, where i is the index of the element (i = 0, 1, 2, ..., N), Write an algorithm based on Greedy technique with a Complexity O(nlogn).

Input Format:

First line specifies the number of elements-n

The next n lines contain the array elements.

Output Format:

Maximum Array Sum to be printed.

Sample Input:

5

2 5 3 4 0

Sample output:

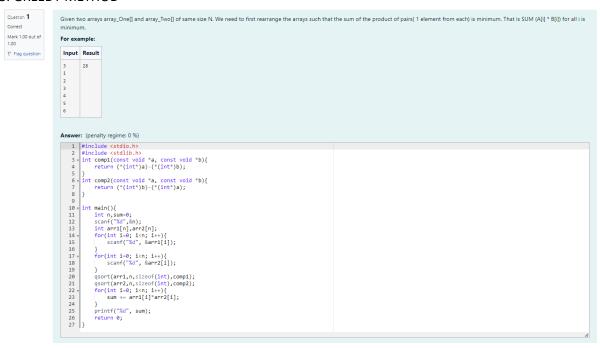
40
```

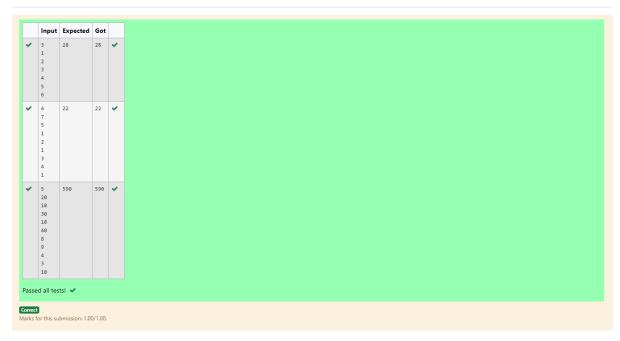
```
Answer: (penalty regime: 0 %)

| ##include settion.h>
| #include s
```

	Input	Expected	Got			
				~		
ľ	5 2	40	40	•		
	5					
	3					
	4					
	9					
•	10	191	191	•		
	2					
	2 2					
	4					
	4					
	3					
	3					
	5					
	5					
	5					
~	2	45	45	~		
	45					
	3					
Pas	Passed all tests! ✓					
	Great					

5. GREEDY METHOD





DIVIDE AND CONQUER

```
Question 1
Correct
Mark 1.00 out of 1.00

Figure Flag question
```

```
Problem Statement
Give an array of 1s and 6s 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 in Size of array
Output Format
First Line Contains Integer - Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

1  | Finc Line Contains Integer - Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

1  | Finc Line Contains Integer - Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

1  | Finc Line Contains Integer - Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

1  | Finc Line Contains Integer - Number of zeroes present in the given array.

4  | Finc Line Contains Integer - Number of zeroes present in the given array.

4  | Finc Line Contains Integer - Number of zeroes present in the given array.

5  | Finc Line Contains Integer - Number of zeroes in the given array.

6  | Finc Line Contains Integer - Number of zeroes in the given array.

6  | Finc Line Contains Integer - Number of zeroes in the given array.

7  | Finc Line Contains Integer - Number of zeroes in the given array.

8  | Finc Line Contains Integer - Number of zeroes in the given array.

8  | Finc Line Contains Integer - Number of zeroes in the given array.

9  | Finc Line Contains Integer - Number of zeroes in the given array.

9  | Finc Line Contains Integer - Number of zeroes in the given array.

1  | Finc Line Contains Integer - Number of zeroes in the given array.

1  | Finc Line Contains Integer - Number of zeroes in the given array.

1  | Finc Line Contains Integer - Number of zeroes in the given array.

1  | Finc Line Contains Integer - Number of zeroes in the given array.

2  | Finc Line Contains Integer - Number of zeroes in the given array.

2  | Finc Line Contains Integer - Number of zeroes in the given array.

2  | Finc Line Contains Integer - Number of zeroes in the given array.

2
```



2. DIVIDE AND CONQUER

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 [n / 2] 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:

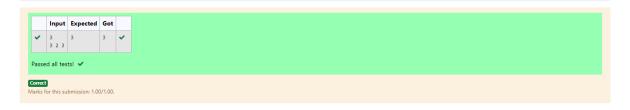
- n == nums.length
 1 <= n <= 5 * 10⁴
 -2³¹ <= nums[i] <= 2³¹ 1

For example:

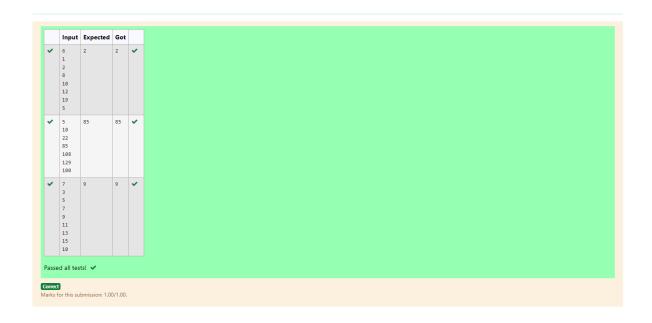
Input	Result
3	3
3 2 3	
7	2
2 2 1 1 1 2 2	

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    int major(int a[],int left,int right);
 2
    int count(int a[],int left,int right,int n);
int major(int a[],int left,int right)
 4
 5 1
 6
         if(left==right)
 7
         {
 8
             return a[left];
 9
10
         int mid=(left+right)/2;
11
         int lm=major(a,left,mid);
12
         int rm=major(a,mid+1,right);
13
         if(lm==rm)
14
         {
15
             return lm;
16
         int lc=count(a,left,right,lm);
17
18
         int rc=count(a,left,right,rm);
19
         return(lc>rc) ? lm:rm;
20
21
    int count(int a[],int left,int right,int n)
22
23 1
24
         int c=0;
25
         for(int i=left;i<=right;i++)</pre>
26
27
            if(a[i]==n)
28
            {
29
                 C++;
30
31
32
33
     return c;
34
35
    int main(){
         int n;
scanf("%d",&n);
36
37
38
         int a[n];
39
         for(int i=0;i<n;i++)</pre>
40
             scanf("%d",&a[i]);
41
42
43
44
         int maj=major(a,0,n-1);
         printf("%d",maj);
45
46
47
48
```



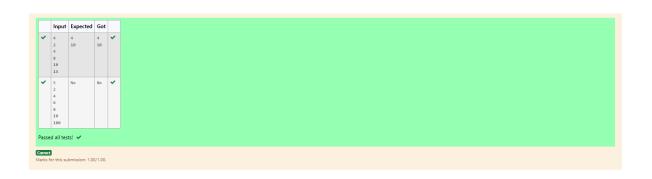
3. DIVIDE AND CONQUER



4.DIVIDE AND CONQUER

```
Problem Statement:
General Statement:
General Statement:
General Statement:
They aurrior

Prop aurri
```



5. DIVIDE AND CONQUER

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

