



HARINI R 2024-IT

**L2****Started on** Wednesday, 6 August 2025, 10:16 PM**State** Finished**Completed on** Wednesday, 6 August 2025, 10:29 PM**Time taken** 12 mins 58 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question1 CorrectMark1.00outof1.00

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void function (int n)
```

```
{
    inti=1;
```

```
    ints=1;
```

```
    while(s<=n)
    {
        i++;
        s+=i;
    }
}
```

Note:Noneedofcounterincrementfordeclarationsandscanf()andcountvariableprintf()statements.

Input:

ApositiveIntegern

Output:

Printthevalueofthecountervariable

Forexample:

Input	Result
9	12

Answer:(penaltyregime:0%)

```
1  #include<stdio.h>
2
3  voidfunction(intn)
4  {
5      intcount=0;
6      //count++;
7      inti=1;
8      count++;
9      ints=1;
10     count++;
11
12     while(s<=n){
13         count++;
14         i++;
15         count++;
16         s+=i;
17         count++;
18     }
19     count++;
20
21     printf("%d",count);
22 }
23
24 intmain()
25 {
26     intnum;
27     scanf("%d",&num);
28     function(num);
29 }
30
```

	Input	Expected	Got	
✓	9	12	12	✓

	Input	Expected	Got	
✓	4	9	9	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)

RESULT:

For input **n = 9**, the output (counter value) is **12**.
Hence, the **time complexity** of the algorithm is **O(n)**.



HARINI R 2024-IT



L2

Started on Thursday, 7 August 2025, 10:12 AM

State Finished

Completed on Thursday, 7 August 2025, 10:31 AM

Time taken 18 mins 48 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question1 Correct Mark 1.00 out of 1.00

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void func(int n)
{
    if(n==1)
    {
        printf("*");
    }
    else
    {
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                printf("*");
                printf("*");
                break;
            }
        }
    }
}
```

Note: None of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer: (penalty regime: 0%)

```
1  #include<stdio.h>
2
3  void func(int n){
4      int count=0;
5
6      if(n==1){
7          count++;
8          printf("*");
9      }
10     else{
11         count++;
12         for(int i=1;i<=n;i++)
13         {
14             count++;
15             for(int j=1;j<=n;j++)
16             {
17                 count++;
18
19                 count++;
20
21                 count++;
22                 break;
23             }
24             count++;
25         }
26         count++;
27     }
28     printf("%d", count);
29 }
30
31
32 int main(){
33     int num;
34     scanf("%d",&num);
35     func(num);
```

```
36  
37}
```

	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	5002	5002	✓
✓	143	717	717	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

RESULT:

For input $n = 3$, the output (counter value) is **16**.
Hence, the **time complexity** of the algorithm is **$O(n^2)$** .



HARINI R 2024-IT



L2

Started on Wednesday, 6 August 2025, 9:53 PM

State Finished

Completed on Wednesday, 6 August 2025, 10:31 PM

Time taken 37 mins 33 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question1 CorrectMark1.00outof1.00

Convert the following algorithm into a program and find its time complexity using counter method.

```
Factor(num) {
{
    for(i=1;i<=num;++i)
    {
        if(num%i==0)
        {
            printf("%d",i);
        }
    }
}
```

Note:Noneedofcounterincrementfordeclarationsandscanf()andcountervariableprintf()statement.

Input:

ApositiveIntegern

Output:

Printthevalueofthecountervariable

Answer:

```
1  #include<stdio.h>
2
3  voidFactor(intnum){
4      intcount=0;
5      //count++;
6      for(inti=1;i<=num;++i){
7          count++;
8          if(num%i==0){
9              count++;
10             //printf("%d",i);
11         }
12         count++;
13     }
14     count++;
15     printf("%d",count);
16 }
17
18 intmain(){
19     intn;
20     scanf("%d",&n);
21     Factor(n);
22 }
23
24
25
```

	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	4	12	12	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

RESULT:

For input $n = 5$, the output (counter value) is **12**.
Hence, the **time complexity** of the algorithm is **$O(n)$** .

[BacktoCourse](#)



HARINI R 2024-IT

**L2****Started on** Wednesday, 6 August 2025, 10:20 PM**State** Finished**Completed on** Wednesday, 6 August 2025, 10:32 PM**Time taken** 12 mins 6 secs**Marks** 1.00/1.00**Grade** 10.00 out of 10.00 (100%)

Question1 CorrectMark1.00outof1.00

Convert the following algorithm into a program and find its time

complexity using counter method.

```
void function(int n)
{
    int c = 0;
    for (int i = n/2; i < n; i++)
        for (int j = 1; j < n; j = 2*j)
            for (int k = 1; k < n; k = k * 2)
                c++;
}
```

Note: Noneed of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1  #include<stdio.h>
2
3  void function(int n){
4      int count=0;
5      count++;
6      for (int i = n/2; i < n; i++){
7          count++;
8          for (int j = 1; j < n; j = 2*j){
9              count++;
10             for (int k = 1; k < n; k = k*2){
11                 count++;
12                 count++;
13             }
14             count++;
15         }
16         count++;
17     }
18     count++;
19
20     printf("%d", count);
21 }
22
23 int main(){
24     int num;
25     scanf("%d", &num);
26     function(num);
27 }
```

	Input	Expected	Got	
✓	4	30	30	✓
✓	10	212	212	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

RESULT:

For input $n = 4$, the output (counter value) is **30**.
Hence, the **time complexity** of the algorithm is **$O(n^3)$** .

[BacktoCourse](#)



HARINI R 2024-IT



L2

Started on Wednesday, 6 August 2025, 10:33 PM

State Finished

Completed on Wednesday, 6 August 2025, 10:34 PM

Time taken 1 min 4 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question1 CorrectMark1.00outof1.00

Convert the following algorithm into a program and find its time complexity using counter method.

```
void reverse(int n)
{
    int rev = 0, remainder;
    while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n /= 10;
    }
    print(rev);
}
```

Note: Noneed of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1  #include<stdio.h>
2
3  void reverse(int n){
4      int count=0;
5      count++;
6      int rev=0, remainder;
7      count++;
8      while(n!=0){
9          count++;
10         remainder=n%10;
11         count++;
12         rev=rev*10+remainder;
13         count++;
14         n/=10;
15         count++;
16     }
17     count++;
18     //print(rev);
19
20     printf("%d", count);
21 }
22
23 int main(){
24     int num;
25     scanf("%d",&num);
26     reverse(num);
27 }
```

	Input	Expected	Got	
✓	12	11	11	✓
✓	1234	19	19	✓

Passed all tests! ✓

Correct

RESULT:

For input **n = 10**, the output (counter value) is **18**.
Hence, the **time complexity** of the algorithm is **$O(\log n)$** .

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)



HARINI R 2024-IT



L2

StartedonThursday,18September 2025,10:26 AM

StateFinished

CompletedonThursday,18September2025,10:39AM

Timetaken12mins16secs

Marks1.00/1.00

Grade10.00outof10.00(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 contain 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 countZeroes(int arr[], int low, int high){
4      if(low > high){
5          return 0;
6      }
7      if(low == high){
8          return arr[low] == 0 ? 1 : 0;
9      }
10     int mid = (low + high) / 2;
11     int leftCount = countZeroes(arr, low, mid);
12     int rightCount = countZeroes(arr, mid + 1, high);
13     return leftCount + rightCount;
14 }
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 result = countZeroes(arr, 0, m - 1);
24     printf("%d\n", result);
25     return 0;
26 }
27

```

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

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

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

RESULT:

The program successfully reads the number of boys, girls, and students, and displays the total number of students who got grade 'A'.

[BacktoCourse](#)



HARINI R 2024-IT



L2

Started on Thursday, 18 September 2025, 10:39 AM

State Finished

Completed on Thursday, 18 September 2025, 10:52 AM

Time taken 13 mins 15 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question1 CorrectMark1.00outof1.00

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.

Example1:

Input: `nums = [3, 2, 3]`

Output: 3

Example2:

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
3 323	3
7 2211122	2

Answer:(penaltyregime:0%)

```

1  #include<stdio.h>
2
3  int majorityElement(int nums[], int n){
4      int count=0, candidate=nums[0];
5      for(int i=0; i<n; i++){
6          if(count==0){
7              candidate=nums[i];
8          }
9          count+=(nums[i]==candidate)?1:-1;
10     }
11     return candidate;
12 }
13
14 int main(){
15     int n;
16     scanf("%d",&n);
17     int nums[n];
18     for(int i=0; i<n; i++){
19         scanf("%d",&nums[i]);
20     }
21     printf("%d\n",majorityElement(nums,n));
22     return 0;
23 }
24

```

//

	Input	Expected	Got	
✓	3 323	3	3	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

BacktoCourse

RESULT:

The program successfully finds and displays the majority element in the given array.



HARINI R 2024-IT



L2

Started on Thursday, 18 September 2025, 11:16 AM

State Finished

Completed on Saturday, 18 October 2025, 9:38 AM

Time taken 29 days 22 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 the array smaller than or equal to x . Write a divide and conquer algorithm to find floor of x .

Input Format

First line contains integer n – Size of array. Next n lines contain 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) return -1;
5      if(x>=arr[high]) return arr[high];
6      int mid=(low+high)/2;
7      if(arr[mid]==x) return arr[mid];
8      if(mid>0&&arr[mid-1]<=x&&x<arr[mid]) return arr[mid-1];
9      if(x<arr[mid]) return findFloor(arr, low, mid-1, x);
10     return findFloor(arr, mid+1, high, x);
11 }
12
13 int main(){
14     int n, x;
15     scanf("%d", &n);
16     int arr[n];
17     for(int i=0; i<n; i++){
18         scanf("%d", &arr[i]);
19     }
20     scanf("%d", &x);
21     int result=findFloor(arr, 0, n-1, x);
22     printf("%d\n", result);
23     return 0;
24 }
25
26

```

	Input	Expected	Got	
✓	6 1 2 8 10 12 19 5	2	2	✓
✓	5 10 22 85 108 129 100	85	85	✓

	Input	Expected	Got	
✓	7	9	9	✓
	3			
	5			
	7			
	9			
	11			
	13			
	15			
	10			

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

BacktoCourse

RESULT:

The program successfully finds and displays the floor value of a given key in a sorted array using the divide-and-conquer method.



HARINI R 2024-IT



L2

Started on Saturday, 18 October 2025, 9:34 AM

State Finished

Completed on Saturday, 18 October 2025, 9:34 AM

Time taken 26 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 `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 exists 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 – Element 1

Second Line Contains Integer – Element 2 (Element 1 and Element 2 together sum 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) return 0;
5      int sum = arr[left] + arr[right];
6      if(sum == x){
7          *a = arr[left];
8          *b = arr[right];
9          return 1;
10     } else if(sum < x){
11         return findPair(arr, left+1, right, x, a, b);
12     } else{
13         return findPair(arr, left, right-1, x, a, b);
14     }
15 }
16
17 int main(){
18     int n, x;
19     scanf("%d", &n);
20     int arr[n];
21     for(int i=0; i<n; i++){
22         scanf("%d", &arr[i]);
23     }
24     scanf("%d", &x);
25     int a, b;
26     if(findPair(arr, 0, n-1, x, &a, &b)){
27         printf("%d\n%d\n", a, b);
28     } else{
29         printf("No\n");
30     }
31     return 0;
32 }
33

```

	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](#)

RESULT:

The program successfully finds and displays the pair of elements whose sum equals the given target value using the divide-and-conquer approach.



HARINI R 2024-IT



L2

StartedonSaturday,18October2025,9:36AM

StateFinished

CompletedonSaturday,18October2025,9:37AM

Timetaken40secs

Marks1.00/1.00

Grade10.00outof10.00(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 number of elements in the list - n. The next n lines contain the elements.

Output:

Sorted list of elements

For example:

Input	Result
5 6734129878	1234677898

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 partition(int arr[],int low,int high){
10     int pivot=arr[high];
11     int i=low-1;
12     for(int j=low;j<high;j++){
13         if(arr[j]<=pivot){
14             i++;
15             swap(&arr[i],&arr[j]);
16         }
17     }
18     swap(&arr[i+1],&arr[high]);
19     return i+1;
20 }
21
22 void quickSort(int arr[],int low,int high){
23     if(low<high){
24         int pi=partition(arr,low,high);
25         quickSort(arr,low,pi-1);
26         quickSort(arr,pi+1,high);
27     }
28 }
29
30 int main(){
31     int n;
32     scanf("%d",&n);
33     int arr[n];
34     for(int i=0;i<n;i++){
35         scanf("%d",&arr[i]);
36     }
37     quickSort(arr,0,n-1);
38     for(int i=0;i<n;i++){
39         printf("%d",arr[i]);
40     }
41     printf("\n");
42     return 0;
43 }
44

```

	Input	Expected	Got	
✓	5 6734129878	1234677898	1234677898	✓
✓	10 15678903256111090114	11011325656789090114	11011325656789090114	✓
✓	12 987654321101190	123456789101190	123456789101190	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)

RESULT:

The program successfully implements the **Quick Sort algorithm**, sorting the array elements in ascending order.



HARINI R 2024-IT



L2

Started on Sunday, 31 August 2025, 11:10 AM

State Finished

Completed on Sunday, 31 August 2025, 11:13 AM

Time taken 3 mins 10 secs

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question1 CorrectMark1.00outof1.00

WriteaprogramtotakevalueVandwewanttomakechangeforVRs,andwehaveinfinitesupplyofeachofthedenominationsin Indiancurrency,i.e.,wehaveinfinitesupplyof{1,2,5,10,20,50,100,500,1000}valuedcoins/notes,whatisthemimum numberofcoinsand/ornotesneededtomakethechange.

InputFormat:

Take an integer from stdin.

OutputFormat:

printtheintegerwhichischangeofthenumber. Example

Input :

64

Output:

4

Explanaton:

Weneeda50Rsnoteanda10Rsnoteandtwo2rupeecoins.

Answer:(penaltyregime:0%)

```

1  #include<stdio.h>int
2  main() {
3      intV,count=0,i;
4      intdenom[]={1000,500,100,50,20,10,5,2,1};
5      scanf("%d",&V);
6      for(i=0;i<9;i++){
7          count+=V/denom[i];
8          V%=denom[i];
9      }
10     printf("%d",count);
11     return 0;
12 }
13

```

	Input	Expected	Got	
	49	5	5	

Passedalltests!

Correct

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)

RESULT:

The program successfully calculates the **minimum number of coins and/or notes** required to make change for a given amount.



HARINI R 2024-IT



L2

Startedon	Sunday,31August2025,11:14 AM
State	Finished
Completedon	Sunday,31August2025,11:16AM
Timetaken	1min54secs
Marks	1.00/1.00
Grade	10.00outof10.00(100%)

Question 1 Correct Mark 1.00 out of 1.00

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 $s[j] \geq g[i]$, 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.

Example 1:**Input:**

```
3
12 3
2
1 1
```

Output:

```
1
```

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:

$1 \leq g.length \leq 3 \times 10^4$

$0 \leq s.length \leq 3 \times 10^4$

$1 \leq g[i], s[j] \leq 2^{31} - 1$

Answer: (penalty regime: 0%)

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int cmp(const void*a, const void*b){
5      return (*(int*)a - *(int*)b);
6  }
7
8  int main(){
9      int n, m, i, j, result=0;
10     scanf("%d", &n);
11     int g[n];
12     for(i=0; i<n; i++) scanf("%d", &g[i]);
13     scanf("%d", &m);
14     int s[m];
15     for(i=0; i<m; i++) scanf("%d", &s[i]);
16     qsort(g, n, sizeof(int), cmp);
17     qsort(s, m, sizeof(int), cmp);
18     i=0; j=0;
19     while(i<n && j<m){
20         if(s[j] >= g[i]){
21             result++;
22             i++;
23             j++;
24         } else{
25             j++;
26         }
27     }
28     printf("%d", result);
29     return 0;
30 }
31
```

//

	Input	Expected	Got	
✓	2	2	2	✓
	12			
	3			
	123			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

RESULT:

The program successfully computes the **maximum number of children** who can be satisfied with the given cookies.



HARINI R 2024-IT



L2

Startedon	Sunday,31August2025,11:16 AM
State	Finished
Completedon	Sunday,31August2025,2:11PM
Timetaken	2hours55mins
Marks	1.00/1.00
Grade	10.00outof10.00(100%)

Question 1 Correct Mark 1.00 out of 1.00

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 $3^i \cdot c$ 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 $(3^0 \cdot 1) + (3^1 \cdot 3) + (3^2 \cdot 2) = 1 + 9 + 18 = 28$.

But this is not the minimum, 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 Format

First line contains the number of burgers

Second line contains calories of each burger which is space-separated integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

```
3
5107
```

Sample Output

```
76
```

For example:

Test	Input	Result
TestCase1	3 132	18

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2  #include<math.h>
3
4  int main(){
5      int n;
6      scanf("%d",&n);
7      int a[n];
8      for(int i=0;i<n;i++){
9          scanf("%d",&a[i]);
10     }
11
12     for(int i=0;i<n;i++){
13         for(int j=0;j<n;j++){
14             if(a[i]>a[j]){
15                 int t=a[i];
16                 a[i]=a[j];
17                 a[j]=t;
18             }
19         }
20     }
21
22     int to=0,s=0;
23     for(int j=0;j<n;j++){
24         to=(pow(n,j))*(a[j]);
25         s+=to;
26     }
27     printf("%d",s);
28     return 0;

```

29} |

	Test	Input	Expected	Got	
✓	TestCase1	3 132	18	18	✓
✓	TestCase2	4 7496	389	389	✓
✓	TestCase3	3 5107	76	76	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)**RESULT:**

The program successfully calculates the **minimum number of calories burned** based on the order of burgers eaten.



HARINI R 2024-IT



L2

Startedon	Sunday,31August2025,11:18 AM
State	Finished
Completedon	Sunday,31August2025,11:19AM
Timetaken	55secs
Marks	1.00/1.00
Grade	10.00outof10.00(100%)

Question 1 Correct Mark 1.00 out of 1.00

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

Input Format:

First line specifies the number of elements n .

Then 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%)

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int cmp(const void*a, const void*b){
5      return (*(int*)a) - (*(int*)b);
6  }
7
8  int main(){
9      int n, i;
10     scanf("%d", &n);
11     int arr[n];
12     for(i=0; i<n; i++) scanf("%d", &arr[i]);
13     qsort(arr, n, sizeof(int), cmp);
14     long long result=0;
15     for(i=0; i<n; i++){
16         result += (long long) arr[i] * i;
17     }
18     printf("%lld", result);
19     return 0;
20 }
21

```

	Input	Expected	Got	
✓	5	40	40	✓
	2			
	5			
	3			
	4			
	0			

	Input	Expected	Got	
✓	10 2 2 2 4 4 3 3 5 5 5	191	191	✓
✓	2 45 3	45	45	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

BacktoCourse

RESULT:

The program successfully finds the **maximum sum of $i \times arr[i]$** using a Greedy algorithm approach.



HARINI R 2024-IT



L2

Startedon	Sunday,31August2025,11:19 AM
State	Finished
Completedon	Sunday,31August2025,11:21AM
Timetaken	1min10secs
Marks	1.00/1.00
Grade	10.00outof10.00(100%)

Question 1 Correct Mark 1.00 out of 1.00

Given two arrays `array_One[]` and `array_Two[]` of same size `N`. We need to first rearrange the array such that the sum of the product of pairs (1 element from each) is minimum. That is $\text{SUM} (A[i] * B[i])$ for all `i` is minimum.

For example:

Input	Result
3	28
1	
2	
3	
4	
5	
6	

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int cmpAsc(const void*a, const void*b){
5      return (*(int*)a) - (*(int*)b);
6  }
7
8  int cmpDesc(const void*a, const void*b){
9      return (*(int*)b) - (*(int*)a);
10 }
11
12 int main(){
13     int n, i;
14     scanf("%d", &n);
15     int A[n], B[n];
16     for(i=0; i<n; i++) scanf("%d", &A[i]);
17     for(i=0; i<n; i++) scanf("%d", &B[i]);
18     qsort(A, n, sizeof(int), cmpAsc);
19     qsort(B, n, sizeof(int), cmpDesc);
20     long long result=0;
21     for(i=0; i<n; i++){
22         result += (long long)A[i]*B[i];
23     }
24     printf("%lld", result);
25     return 0;
26 }
27

```

	Input	Expected	Got	
✓	3	28	28	✓
	1			
	2			
	3			
	4			
	5			
	6			

	Input	Expected	Got	
✓	4 7 5 1 2 1 3 4 1	22	22	✓
✓	5 20 10 30 10 40 8 9 4 3 10	590	590	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

RESULT:

The program successfully computes the **minimum product** of k elements in an array using a **Greedy approach**.



HARINI R 2024-IT



L2

Started on Friday, 31 October 2025, 6:59 AM

State Finished

Completed on Friday, 31 October 2025, 7:02 AM

Time taken 3 mins 26 secs

Grade 10.00 out of 10.00 (100%)

Question1 CorrectMark10.00outof10.00**PlayingwithNumbers:**

RamandSitaareplayingwithnumbersbygivingpuzzlestoeachother.NowitwasRamterm,sohegaveSitaapositiveinteger'n' andtwonumbers1and3.Heaskedhertofindthepossiblewaysbywhichthenumberncanberepresentedusing1and3.Write any efficient algorithm to find the possible ways.

Example1:*Input:*6*Output:*6*Explanation:* Thereare6waysto6representnumberwith1and3

1+1+1+1+1+1

3+3

1+1+1+3

1+1+3+1

1+3+1+1

3+1+1+1

Input Format

FirstLinecontainsthenumbere

OutputFormat**Print:**The number of possible ways 'n' can be represented using 1 and 3

SampleInput

6

SampleOutput

6

Answer:(penaltyregime:0%)

```

1  #include<stdio.h>
2
3  intmain(){
4      intn;
5      scanf("%d",&n);
6      longlongdp[n+1];
7      dp[0]=1;
8      for(inti=1;i<=n;i++){
9          dp[i]=dp[i-1];
10         if(i>=3)dp[i]+=dp[i-3];
11     }
12     printf("%lld",dp[n]);
13     return0;
14 }
15

```

	Input	Expected	Got	
✓	6	6	6	✓

	Input	Expected	Got	
✓	25	8641	8641	✓
✓	100	24382819596721629	24382819596721629	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)

RESULT:

The program successfully computes the **number of possible ways** to represent a given integer n as a sum of 1s and 3s using **Dynamic Programming**.



HARINI R 2024-IT



L2

Started on Saturday, 18 October 2025, 9:40 AM

State Finished

Completed on Saturday, 18 October 2025, 9:41 AM

Time taken 5 mins 30 secs

Grade 10.00 out of 10.00 (100%)

Question 1 Correct Mark 10.00 out of 10.00**Playing with Chessboard:**

Ram is given with an $n \times n$ chessboard with each cell with a monetary value. Ram stands at the $(0,0)$, that is the position of the top left white rook. He is being given a task to reach the bottom right black rook position $(n-1, n-1)$ constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help Ram to achieve it by providing an efficient DP algorithm.

Example:**Input**

3

124

234

871

Output:

19

Explanation:

Totally there will be 6 paths among that the optimal is Optimal path value: $1+2+8+7+1=19$

Input Format

First line contains the integer n

Then next n lines contain the $n \times n$ chessboard values

Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2
3  int max(int a, int b){
4      return a>b?a:b;
5  }
6
7  int main(){
8      int n;
9      scanf("%d",&n);
10     int board[n][n];
11     for(int i=0;i<n;i++){
12         for(int j=0;j<n;j++){
13             scanf("%d",&board[i][j]);
14         }
15     }
16     int dp[n][n];
17     dp[0][0]=board[0][0];
18     for(int i=1;i<n;i++){
19         dp[i][0]=dp[i-1][0]+board[i][0];
20         dp[0][i]=dp[0][i-1]+board[0][i];
21     }
22     for(int i=1;i<n;i++){
23         for(int j=1;j<n;j++){
24             dp[i][j]=max(dp[i-1][j],dp[i][j-1])+board[i][j];
25         }
26     }
27     printf("%d\n",dp[n-1][n-1]);
28     return 0;
29 }
30

```

	Input	Expected	Got	
✓	3 124 234 871	19	19	✓
✓	3 131 151 421	12	12	✓
✓	4 1134 1578 2346 1690	28	28	✓

Passedalltests! ✓

Correct

Marksforthissubmission:10.00/10.00.

[BacktoCourse](#)

RESULT:

The program successfully computes the **number of unique paths** on a chessboard using **Dynamic Programming**.



Started on Saturday, 18 October 2025, 9:39 AM
State Finished
Completed on Saturday, 18 October 2025, 9:40 AM
Time taken 6 mins 7 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 two strings find the length of the common longest subsequence (need not be contiguous) between the two.

Example:

s1:ggtabe

s2:tgatab

s1		a		g		g		t		a		b
s2		g		x		t		x		a		y
												b

The length is 4

Solve it using Dynamic Programming

For example:

Input	Result
aab	2
azb	

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2  #include<string.h>
3
4  int max(int a, int b){
5      return a>b?a:b;
6  }
7
8  int main(){
9      char s1[1000], s2[1000];
10     scanf("%s", s1);
11     scanf("%s", s2);
12     int n=strlen(s1);
13     int m=strlen(s2);
14     int dp[n+1][m+1];
15     for(int i=0; i<=n; i++){
16         for(int j=0; j<=m; j++){
17             if(i==0 || j==0)
18                 dp[i][j]=0;
19             else if(s1[i-1]==s2[j-1])
20                 dp[i][j]=dp[i-1][j-1]+1;
21             else
22                 dp[i][j]=max(dp[i-1][j], dp[i][j-1]);
23         }
24     }
25     printf("%d\n", dp[n][m]);
26     return 0;
27 }
28

```

	Input	Expected	Got	
✓	aab	2	2	✓
	azb			
✓	ABCD	4	4	✓
	ABCD			

Passed all tests! 

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

Result:

- The final value $dp[m][n]$ gives the **length of the Longest Common Subsequence**.



HARINI R 2024-IT



L2

StartedonSaturday,18October2025,9:38AM

StateFinished

CompletedonSaturday,18October2025,9:39AM

Timetaken5min8secs

Marks1.00/1.00

Grade10.00outof10.00(100%)

Question1 CorrectMark1.00outof1.00

Problemstatement:

FindthelengthoftheLongestNon-decreasingSubsequenceinagivenSequence.

Eg:

Input:9

Sequence: [-1,3,4,5,2,2,2,3]

thesubsequenceis [-1,2,2,2,2,3] Output:6

Answer:(penaltyregime:0%)

```

1  #include<stdio.h>
2
3  intmax(inta,intb){
4      returna>b?a:b;
5  }
6
7  intmain(){
8      intn;
9      scanf("%d",&n);
10     inta[n],dp[n];
11     for(inti=0;i<n;i++){
12         scanf("%d",&a[i]);
13         dp[i]=1;
14     }
15     for(inti=1;i<n;i++){
16         for(intj=0;j<i;j++){
17             if(a[i]>=a[j]){
18                 dp[i]=max(dp[i],dp[j]+1);
19             }
20         }
21     }
22     intmaxLen=0;
23     for(inti=0;i<n;i++){
24         if(dp[i]>maxLen)maxLen=dp[i];
25     }
26     printf("%d\n",maxLen);
27     return0;
28 }
29

```

	Input	Expected	Got	
✓	9 -134522223	6	6	✓
✓	7 1224576	6	6	✓

Passedalltests!

**Correct**

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)



HARINI R 2024-IT

**L2****Started on** Saturday, 18 October 2025, 9:28AM**State** Finished**Completed on** Saturday, 18 October 2025, 9:31AM**Time taken** 2mins 19secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

Question 1 Correct Mark 1.00 out of 1.00

FindDuplicateinArray.

Givenareadonlyarrayofnintegersbetween1andn,findonenumeratorthatrepeats. InputFormat:

FirstLine-Numberofelements n

Lines - n Elements

OutputFormat:

Elementx-Thatisrepeated

Forexample:

Input	Result
5 11234	1

Answer:(penaltyregime:0%)

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  intmain(){
5      intn;
6      scanf("%d",&n);
7      inta[n];
8      for(inti=0;i<n;i++){
9          scanf("%d",&a[i]);
10     }
11     for(inti=0;i<n;i++){
12         intindex=abs(a[i])-1;
13         if(a[index]<0){
14             printf("%d\n",abs(a[i]));
15             return0;
16         }
17         a[index]=-a[index];
18     }
19     return0;
20 }
```

	Input	Expected	Got	
✓	11 109765123847	7	7	✓
✓	5 12344	4	4	✓
✓	5 11234	1	1	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)

RESULT:

The program successfully identifies and prints duplicate elements in an array using **$O(n^2)$ time** and **$O(1)$ space** complexity.



HARINI R 2024-IT

**L2****Started on** Saturday, 18 October 2025, 9:31AM**State** Finished**Completed on** Saturday, 18 October 2025, 9:32AM**Time taken** 37secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

Question 1 Correct Mark 1.00 out of 1.00

Find Duplicate in Array.

Given an array of n integers between 1 and n , find one number that repeats. Input Format:First Line- Number of elements n Lines - n Elements

Output Format:

Element x - That is repeated**For example:**

Input	Result
5 1 1 2 3 4	1

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main(){
5      int n;
6      scanf("%d",&n);
7      int a[n];
8      for(int i=0;i<n;i++){
9          scanf("%d",&a[i]);
10     }
11     for(int i=0;i<n;i++){
12         int index=abs(a[i])-1;
13         if(a[index]<0){
14             printf("%d\n",abs(a[i]));
15             return 0;
16         }
17         a[index]=-a[index];
18     }
19     return 0;
20 }
```

	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

RESULT:

The program efficiently finds duplicate elements in an array using **$O(n)$ time** and **$O(1)$ space** complexity.



HARINI R 2024-IT



L2

Started on Saturday, 18 October 2025, 9:32 AM

State Finished

Completed on Saturday, 18 October 2025, 9:32 AM

Time taken 3 mins 8 secs

Marks 1.00/1.00

Grade 30.00 out of 30.00 (100%)

Question 1 Correct Mark 1.00 out of 1.00

Find the intersection of two sorted arrays. OR

in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays. Input Format

· The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second

arrayOutputFormat

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 27 10 15 57 24 6

Output:

10 57

Input:

1

6 12 34 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 27 10 15 57 24 6	10 57

Answer: (penalty regime: 0%)

```

1 #include<stdio.h>
2
3 int main(){
4     int T;
5     scanf("%d",&T);
6     while(T--){
7         int N1,N2;
8         scanf("%d",&N1);
9         int a[N1];
10        for(int i=0;i<N1;i++){
11            scanf("%d",&a[i]);
12        }
13        scanf("%d",&N2);
14        int b[N2];
15        for(int i=0;i<N2;i++){
16            scanf("%d",&b[i]);
17        }
18        int i=0,j=0;
19        while(i<N1&&j<N2){
20            if(a[i]<b[j]){
21                i++;
22            }elseif(a[i]>b[j]){
23                j++;
24            }else{

```

```
25         printf("%d",a[i]);
26         i++;
27         j++;
28     }
29 }
30 printf("\n");
31 }
32 return 0;
33 }
34 }
```

	Input	Expected	Got	
✓	1 310 17 57 6 27 10 1557 246	1057	1057	✓
✓	1 61 23 45 2 16	16	16	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

RESULT:

The program successfully finds the intersection of two sorted arrays using **O(m × n) time** and **O(1) space**.



HARINI R 2024-IT



L2

Started onSaturday, 18 October 2025, 9:33AM

StateFinished

Completed onSaturday, 18 October 2025, 9:33AM

Time taken5mins60secs

Marks1.00/1.00

Grade30.00outof30.00(100%)

Question 1 Correct Mark 1.00 out of 1.00

Find the intersection of two sorted arrays. OR

in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays. Input Format

· The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second

array Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 27 10 15 57 24 6

Output:

10 57

Input:

1

6 12 34 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 27 10 15 57 24 6	10 57

Answer: (penalty regime: 0%)

```

1 #include<stdio.h>
2
3 int main(){
4     int T;
5     scanf("%d",&T);
6     while(T--){
7         int N1,N2;
8         scanf("%d",&N1);
9         int a[N1];
10         for(int i=0;i<N1;i++){
11             scanf("%d",&a[i]);
12         }
13         scanf("%d",&N2);
14         int b[N2];
15         for(int i=0;i<N2;i++){
16             scanf("%d",&b[i]);
17         }
18         int i=0,j=0;
19         while(i<N1&&j<N2){
20             if(a[i]<b[j]){
21                 i++;
22             }elseif(a[i]>b[j]){
23                 j++;
24             }else{

```

```

25         printf("%d",a[i]);
26         i++;
27         j++;
28     }
29 }
30 printf("\n");
31 }
32 return 0;
33 }
34

```

	Input	Expected	Got	
✓	1 310 17 57 6 27 10 1557 246	1057	1057	✓
✓	1 61 23 45 2 16	16	16	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

RESULT:

The program successfully finds the intersection of two sorted arrays in **O(m + n) time** and **O(1) space**.

**Started on** Saturday, 18 October 2025, 9:35AM**State** Finished**Completed on** Saturday, 18 October 2025, 9:35AM**Time taken** 4mins5secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

Question 1 Correct Mark 1.00 out of 1.00

Given an array A of sorted integers and another non-negative integer k , find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

Input Format:

First Line - Number of elements in an array Next n

Lines - N elements in the array

k - Non-Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase: YES

as $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3 135 4	1

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2
3  int main(){
4      int n,k;
5      scanf("%d",&n);
6      int a[n];
7      for(int i=0;i<n;i++){
8          scanf("%d",&a[i]);
9      }
10     scanf("%d",&k);
11     int i=0,j=1;
12     while(i<n&&j<n){
13         int diff=a[j]-a[i];
14         if(diff==k&&i!=j){
15             printf("1\n");
16             return 0;
17         }elseif(diff<k){
18             j++;
19         }else{
20             i++;
21             if(i==j)j++;
22         }
23     }
24     printf("0\n");
25     return 0;
26 }
27

```

	Input	Expected	Got	
✓	3 135 4	1	1	✓

	Input	Expected	Got	
✓	10 1468121415202125 1	1	1	✓
✓	10 1235111416242829 0	0	0	✓
✓	10 0237131415202425 10	1	1	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)

RESULT:

The program successfully determines whether there exists a pair of integers in the given array with the specified difference, achieving **$O(n^2)$ time** and **$O(1)$ space complexity**.



HARINI R 2024-IT



L2

StartedonSaturday,18October2025,9:36AM

StateFinished

CompletedonSaturday,18October2025,9:36AM

Timetaken3mins50secs

Marks1.00/1.00

Grade4.00outof4.00(**100%**)

Question 1 Correct Mark 1.00 out of 1.00

Given an array A of sorted integers and another non-negative integer k , find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

Input Format:

First Line - Number of elements in an array Next n

Lines - N elements in the array

k - Non-Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase: YES

as $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3 135 4	1

Answer: (penalty regime: 0%)

```

1  #include<stdio.h>
2
3  int main(){
4      int n,k;
5      scanf("%d",&n);
6      int a[n];
7      for(int i=0;i<n;i++){
8          scanf("%d",&a[i]);
9      }
10     scanf("%d",&k);
11     int i=0,j=1;
12     while(i<n&&j<n){
13         int diff=a[j]-a[i];
14         if(diff==k&&i!=j){
15             printf("1\n");
16             return 0;
17         }elseif(diff<k){
18             j++;
19         }else{
20             i++;
21             if(i==j)j++;
22         }
23     }
24     printf("0\n");
25     return 0;
26 }
27

```

	Input	Expected	Got	
✓	3 135 4	1	1	✓

	Input	Expected	Got	
✓	10 1468121415202125 1	1	1	✓
✓	10 1235111416242829 0	0	0	✓
✓	10 0237131415202425 10	1	1	✓

Passedalltests! ✓

Correct

Marksforthissubmission:1.00/1.00.

[BacktoCourse](#)

RESULT:

The program successfully identifies whether a pair with the given difference exists in an array using **$O(n)$** time and **$O(1)$** space complexity.