CS23331-Design and Analysis of Algorithms-2024 Batch-CSE, IT, AIML & AIDS

Started on	Friday, 1 August 2025, 1:43 PM
State	Finished
Completed on	Friday, 8 August 2025, 2:09 PM
Time taken	7 days
Marks	15.00/15.00
Grade	**100.00** out of 100.00

Quiz navigation

1 2 3 4 5 6 7 8 9
10 11 12 13 14 15

Show one page at a time

Finish review

Question 1

Correct Mark 1.00 out of 1.00 Flag question

Given two numbers, write a C program to swap the given numbers.

For example:

Input	Result
10 20	20 10

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int num1,num2,temp;
    scanf("%d %d",&num1,&num2);
    temp=num1;
    num1=num2;
    num2=temp;
    printf("%d %d",num1,num2);
}
```

	Input	Expected	Got	
✔	10 20	20 10	20 10	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths >= 65

Marks in Physics >= 55

Marks in Chemistry >= 50

Or

Total in all three subjects >= 180

Sample Test Cases

Test Case 1

Input

70 60 80

Output

The candidate is eligible

Test Case 2

Input

50 80 80

Output

The candidate is eligible

Test Case 3

Input

50 60 40

Output

The candidate is not eligible

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int mark1,mark2,mark3;
    scanf("%d %d %d",&mark1,&mark2,&mark3);
    if((mark1>=65 && mark2>=55 && mark3>=50 ) || mark1+mark2+mark3>=180) printf("The candidate is eligible");
```

```
    else printf("The candidate is not eligible");
}
```

	Input	Expected	Got	
✔	70 60 80	The candidate is eligible	The candidate is eligible	✔
✔	50 80 80	The candidate is eligible	The candidate is eligible	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00 Flag question

Malini goes to BestSave hyper market to buy grocery items. BestSave hyper market provides 10% discount on the bill amount B when ever the bill amount B is more than Rs.2000.

The bill amount B is passed as the input to the program. The program must print the final amount A payable by Malini.

Input Format:

The first line denotes the value of B.

Output Format:

The first line contains the value of the final payable amount A.

Example Input/Output 1:

Input:

1900

Output:

1900

Example Input/Output 2:

Input:

3000

Output:

2700

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int amt;
    scanf("%d",&amt);
    if(amt>2000){
        int dis=amt*0.1;
        amt=amt-dis;
        printf("%d",amt);
    }
```

```
    else printf("%d",amt);
}
```

	Input	Expected	Got	
✔	1900	1900	1900	✔
✔	3000	2700	2700	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct Mark 1.00 out of 1.00 Flag question

Baba is very kind to beggars and every day Baba donates half of the amount he has when ever a beggar requests him. The money M left in Baba's hand is passed as the input and the number of beggars B who received the alms are passed as the input. The program must print the money Baba had in the beginning of the day.

Input Format:

The first line denotes the value of M.
The second line denotes the value of B.

Output Format:

Output Format:

The first line denotes the value of money with Baba in the beginning of the day.

Example Input/Output:

Input:

100
2

Output:

400

Explanation:

Baba donated to two beggars. So when he encountered second beggar he had 100*2 = Rs.200 and when he encountered 1st he had 200*2 = Rs.400.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int main(){
    int amt,beg;
    scanf("%d %d",&amt,&beg);
    int x=pow(2,beg);
    printf("%d",amt*x);
}
```

	Input	Expected	Got	
✔	100 2	400	400	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00 Flag question

The CEO of company ABC Inc wanted to encourage the employees coming on time to the office. So he announced that for every consecutive day an employee comes on time in a week (starting from Monday to Saturday), he will be awarded Rs.200 more than the previous day as "Punctuality Incentive". The incentive **I** for the starting day (ie on Monday) is passed as the input to the program. The number of days N an employee came on time consecutively starting from Monday is also passed as the input. The program must calculate and print the "Punctuality Incentive" P of the employee.

Input Format:

The first line denotes the value of **I**.
The second line denotes the value of N.

Output Format:

The first line denotes the value of **P**.

Example Input/Output:

Input:

500
3

Output:

2100

Explanation:

On Monday the employee receives Rs.500, on Tuesday Rs.700, on Wednesday Rs.900

So total = Rs.2100

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int amt,cons;
    scanf("%d %d",&amt,&cons);
    printf("%d",(amt*cons)+(200*cons));
}
```

	Input	Expected	Got	
✔	500 3	2100	2100	✔
✔	100 3	900	900	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00 Flag question

Two numbers M and N are passed as the input. A number X is also passed as the input. The program must print the numbers divisible by X from N to M (inclusive of M and N).

Input Format:

The first line denotes the value of M
The second line denotes the value of N
The third line denotes the value of X

Output Format:

Numbers divisible by X from N to M, with each number separated by a space.

Boundary Conditions:

1 <= M <= 9999999
M < N <= 9999999
1 <= X <= 9999

Example Input/Output 1:

Input:
2
40
7

Output:
35 28 21 14 7

55 28 21 14 7

Example Input/Output 2:

Input:
66
121
11

Output:
121 110 99 88 77 66

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int s,e,x;
    scanf("%d %d %d",&s,&e,&x);
    for(int i=e;i>=s;i--){
        if(i%x==0)printf("%d ",i);
    }
}
```

	Input	Expected	Got	
✔	2 40 7	35 28 21 14 7	35 28 21 14 7	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the quotient and reminder of given integers.

For example:

Input	Result
12 3	4 0

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int i,j;
    scanf("%d %d",&i,&j);
    printf("%d\n%d",(i/j),(i%j));
}
```

	Input	Expected	Got	
✔	12 3	4 0	4 0	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the biggest among the given 3 integers?

For example:

Input	Result
10 20 30	30

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int i,j,k;
    scanf("%d %d %d",&i,&j,&k);
    printf("%d",(i>j)?((i>k)?i:k):(j>k)?j:k);
}
```

	Input	Expected	Got	

✔	10 20 30	30	30	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find whether the given integer is odd or even?

For example:

Input	Result
12	Even
11	Odd

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int x;
    scanf("%d",&x);
    if(x%2==0)printf("Even");
    else printf("Odd");
}
```

	Input	Expected	Got	
✔	12	Even	Even	✔
✔	11	Odd	Odd	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the factorial of given n.

For example:

Input	Result
5	120

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int x,fact=1;
    scanf("%d",&x);
    while(x!=0){
        fact=fact*x;
        x--;
    }
    printf("%d",fact);
}
```

	Input	Expected	Got	
✔	5	120	120	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 11 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the sum first N natural numbers.

For example:

Input	Result
3	6

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int x,sum=0;
    scanf("%d",&x);
    while(x!=0){
        sum+=x;
        x--;
    }
    printf("%d",sum);
}
```

	Input	Expected	Got	
✔	3	6	6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 12 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the Nth term in the fibonacci series.

For example:

Input	Result
0	0
1	1
4	3

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int x;
    scanf("%d",&x);
    int f1=0;int f2=1;
    int f3=1;
    for(int i=0;i<x-1;i++){
        f3=f1+f2;
        f1=f2;
        f2=f3;
    }
    if(x==0)printf("%d",0);
```

```
    else printf("%d",f3);

}
```

	Input	Expected	Got	
✔	0	0	0	✔
✔	1	1	1	✔
✔	4	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 13 | Correct | Mark 1.00 out of 1.00 | Flag question

Write a C program to find the power of integers.

input:

a b

output:

a^b value

For example:

Input	Result
2 5	32

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int main(){
    int x,y;
    scanf("%d %d",&x,&y);
    int ans=pow(x,y);
    printf("%d",ans);
}
```

	Input	Expected	Got	
✔	2 5	32	32	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 14 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find Whether the given integer is prime or not.

For example:

Input	Result
7	Prime

9	No Prime

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int num,flag=1;
    scanf("%d",&num);
    if(num <=1){
        flag=0;
    }
    else{
        for(int i=2;i*i<=num;i++){
            if(num%i==0){
                flag=0;
            }
        }
    }

    if(flag) printf("Prime");
    else printf("No Prime");
}
```

	Input	Expected	Got	
✔	7	Prime	Prime	✔
✔	9	No Prime	No Prime	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question 15 | Correct Mark 1.00 out of 1.00 Flag question

Write a C program to find the reverse of the given integer?

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int num;
    scanf("%d",&num);
    int num2=num;
    int rev=0;
    while(num2!=0){
        rev=rev*10+num2%10;
        num2=num2/10;
    }
    printf("%d",rev);

}
```

	Input	Expected	Got	
✔	123	321	321	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Problem 1: Finding Complexity using Counter Method

Started on	Monday, 3 November 2025, 6:10 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:12 PM
Time taken	1 min 19 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

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

```
void function (int n)
{
    int i= 1;

    int s =1;

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

```
        s += i;
    }
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() s

Input:

A positive Integer n

Output:

Print the value of the counter variable

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int counter=0;
    int i=1;
    counter++;
    int s=1;
    counter++;
    int n;
    scanf("%d",&n);
    while(s<=n)
    {
        counter++;
        i++;
        counter++;
        s+=i;
        counter++;
    }
    counter++;
    printf("%d",counter);

}
```

	Input	Expected	Got	
✔	9	12	12	✔
✔	4	9	9	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 2: Finding Complexity using Counter method

Started on	Monday, 3 November 2025, 6:12 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:13 PM
Time taken	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

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

```
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: No need of counter increment for declarations and scanf() and count variable printf() s

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>

int func(int n){
    int counter=0;
    if(n==1){
        counter++;
        counter++;
        return counter;
    }
    else{
        for(int i=1;i<=n;i++){
            counter++;
            for(int j=1;j<=i;j++){
                counter++;
                counter++;
                counter++;
                counter++;
                break;
            }
        }
    }
    counter++;
    counter++;
    return counter;
}
```

```
int main(){
    int n;
    scanf("%d",&n);
    printf("%d",func(n));
}
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 3: Finding Complexity using Counter Method

Started on	Monday, 3 November 2025, 6:13 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:13 PM
Time taken	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 Flag question

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

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

```
}
```

Note: No need of counter increment for declarations and scanf() and counter variable printf()

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```c
#include<stdio.h>

int main()
{
    int n,i,counter=0;
    scanf("%d",&n);

    for(i=1;i<=n;i++){
        counter++;
        counter++;
        if(n%i==0){
            counter++;
        }
    }
    counter++;

    printf("%d",counter);
}
```

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

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 4: Finding Complexity using Counter Method

Started on	Monday, 3 November 2025, 6:14 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:14 PM
Time taken	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 Flag question

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)
```

Quiz navigation

1

Finish review

```
            c++;
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() s

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```c
#include<stdio.h>
int main()
{
    int n,counter=0;
    scanf("%d",&n);
    counter++;

    for(int i=n/2; i<n;i++){
        counter++;
        for(int j=1;j<n;j=2*j)
        {
            counter++;
            for(int k=1;k<n;k=k*2){
                counter++;
                counter++;
        }
            counter++;
        }
        counter++;
    }
    counter++;
    printf("%d",counter);
}
```

	Input	Expected	Got	
✔	4	30	30	✔
✔	10	212	212	✔

Passed all tests! ✔

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary

Problem 5: Finding Complexity using counter method

Started on	Monday, 3 November 2025, 6:14 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:14 PM
Time taken	18 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

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

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

```
    }
print(rev);
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() s

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```c
#include<stdio.h>
int main()
{
    int rev=0,remainder,counter=0,n;
    scanf("%d",&n);
    counter++;
    counter++;
    while(n!=0)
    {
        counter++;
        remainder=n%10;
        counter++;
        rev=rev*10+remainder;
        counter++;
        n/=10;
        counter++;
    }
    counter++;
    printf("%d",counter);
}
```

	Input	Expected	Got	
✔	12	11	11	✔
✔	1234	19	19	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

1-Number of Zeros in a Given Array

Started on	Monday, 3 November 2025, 6:24 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:25 PM
Time taken	24 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 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.

Quiz navigation

1

Finish review

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>

int dac(int arr[], int low, int high, int n)
{
    if(high<low){
        return 0;
    }
    int mid=(low+high)/2;
    if(arr[mid]==0 && (mid==0|| arr[mid-1]==1)){
        return n-mid;
    }
    else if (arr[mid]==0){
        return dac(arr,low,mid-1,n);
    }
    else{
        return dac(arr,mid+1,high,n);
    }
}

int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }

    int ans=dac(arr,0,n-1,n);
    printf("%d",ans);
}
```

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

	1 1 1 1 1 1 1 1			
✔	8 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 0 0	2	2	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary

2-Majority Element

Started on	Monday, 3 November 2025, 6:25 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:25 PM
Time taken	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 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]
```

Quiz navigation

1

Finish review

Output: 2

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-2^31 <= nums[i] <= 2^31 - 1`

For example:

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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdlib.h>

int cmpfunc(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}

int majorityElement(int* nums, int n) {
    qsort(nums, n, sizeof(int), cmpfunc);
    return nums[n / 2];
}

int main() {
    int n;
    scanf("%d", &n);

    int nums[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }

    printf("%d\n", majorityElement(nums, n));
    return 0;
}
```

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

3-Finding Floor Value

Started on	Monday, 3 November 2025, 6:25 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:26 PM
Time taken	19 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

Quiz navigation

1

Finish review

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findFloor(int arr[], int n, int x) {
    int low = 0, high = n - 1;
    int floorVal = 0;

    while (low <= high) {
        int mid = (low + high) / 2;

        if (arr[mid] == x) {
            return arr[mid];
        }
        else if (arr[mid] < x) {
            floorVal = arr[mid];
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }

    return floorVal;
}

int main() {
    int n, x;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    scanf("%d", &x);

    int floorVal = findFloor(arr, n, x);
    printf("%d\n", floorVal);

    return 0;
}
```

	Input	Expected	Got	
✔	6 1 2	2	2	✔

	8 10 12 19 5			
✔	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.

Finish review

Back to Course

4-Two Elements sum to x

Started on	Monday, 3 November 2025, 6:26 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:26 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 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 %)

```c
#include<stdio.h>

int dac(int arr[],int low, int high, int x, int *a, int *b){
    if(high<low)
        return 0;

    int sum=arr[low]+arr[high];

    if(sum==x){
        *a=arr[low];
        *b=arr[high];
        return 1;
    }
    else if(sum<x){
        return dac(arr,low+1,high,x,a,b);
    }
    else{
        return dac(arr,low,high-1,x,a,b);
    }
}

int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    int x,a,b;
    scanf("%d",&x);
    if(dac(arr,0,n-1,x,&a,&b)){
        printf("%d\n%d",a,b);
    }
    else{
        printf("No");
    }
}
```

	Input	Expected	Got	
✔	4 2 4 8	4 10	4 10	✔

	10 14			
✔	5 2 4 6 8 10 100	No	No	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

5-Implementation of Quick Sort

Started on	Monday, 3 November 2025, 6:26 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:27 PM
Time taken	15 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

Quiz navigation

1

Finish review

5 67 34 12 98 78	12 34 67 78 98

Answer:

```c
#include <stdio.h>
#include <stdlib.h>

int cmpfunc(const void *a, const void *b) {
    return (*(int*)a - *(int*)b);
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    qsort(arr, n, sizeof(int), cmpfunc);

    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}
```

	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 9
✔	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 9

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00

Marks for this submission: 1.00/ 1.00.

Finish review

Back to Course

Data retention summary

1-G-Coin Problem

Started on	Tuesday, 2 September 2025, 2:44 PM
State	Finished
Completed on	Tuesday, 2 September 2025, 2:51 PM
Time taken	7 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

Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input :

64

Quiz navigation

1

Finish review

Output:

4

Explanaton:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int money;
    scanf("%d",&money);
    int count=0;
    int denom[]={1000.500,100,50,20,10,5,2,1};
    int length=sizeof(denom)/sizeof(denom[0]);
    for(int i=0;i<length;i++){
        if(money>=denom[i]){
            count+=money/denom[i];
            money=money%denom[i];
        }
    }
    printf("%d",count);
}
```

	Input	Expected	Got	
✔	49	5	5	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary

2-G-Cookies Problem

Started on	Tuesday, 2 September 2025, 2:51 PM
State	Finished
Completed on	Tuesday, 2 September 2025, 2:55 PM
Time taken	4 mins
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

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.

Example 1:

Input:

3

1 2 3

Quiz navigation

1

Finish review

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 <= g.length <= 3 * 10^4

0 <= s.length <= 3 * 10^4

1 <= g[i], s[j] <= 2^31 - 1

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int child;
    scanf("%d",&child);
    int greed[child];
    for(int i=0;i<child;i++) scanf("%d",&greed[i]);
    int cookies;
    scanf("%d",&cookies);
    int size[cookies];
    for(int i=0;i<cookies;i++){
        scanf("%d",&size[i]);
    }
    int content=0;
    for(int i=0;i<child;i++){
        for(int j=0;j<cookies;j++){
            if(size[j]>=greed[i]){
                content++;
                size[j]=0;
                break;
            }
        }
    }
    printf("%d",content);
}
```

	Input	Expected	Got	
✔	2 1 2 3 1 2 3	2	2	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

3-G-Burger Problem

Started on	Tuesday, 2 September 2025, 2:56 PM
State	Finished
Completed on	Tuesday, 2 September 2025, 3:09 PM
Time taken	13 mins
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

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a distance t
If he has eaten i burgers with c calories each, then he has to run at least $3^i * c$ kilometers to burn out the calories. For example,
burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2) = 1 + 9 +$
But this is not the minimum, so need to try out other orders of consumption and choose the minimum value. Determine the minimum distar
he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm.Apply greedy approach to solve the proble

Input Format

First Line contains the number of burgers

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

Output Format

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

Sample Input

```
3
5 10 7
```

Sample Output

```
76
```

For example:

Test	Input	Result
Test Case 1	3 1 3 2	18

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int comp(const void *a,const void *b){
    return (*(int *)b-*(int *)a);
}

int main(){
    int n;
    scanf("%d",&n);
    int arr1[n];
    for(int i=0;i<n;i++) scanf("%d",&arr1[i]);
    qsort(arr1,n,sizeof(int),comp);
    int km=0;
    for(int i=0;i<n;i++){
        km+=pow(n,i)*arr1[i];
    }
    printf("%d",km);
}
```

	Test	Input	Expected	Got	
✔	Test Case 1	3 1 3 2	18	18	✔
✔	Test Case 2	4 7 4 9 6	389	389	✔
✔	Test Case 3	3 5 10 7	76	76	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

4-G-Array Sum max problem

Started on	Tuesday, 2 September 2025, 3:00 PM
State	Finished
Completed on	Tuesday, 2 September 2025, 3:00 PM
Time taken	19 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 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 %)

```c
#include<stdio.h>
#include<stdlib.h>
int comp(const void *a,const void*b){
    return (*(int *)a-*(int *)b);
}
int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    qsort(arr,n,sizeof(int),comp);
    int sum=0;
    for(int i=0;i<n;i++) sum+=arr[i]*i;
    printf("%d",sum);
}
```

	Input	Expected	Got	
✔	5 2 5 3 4 0	40	40	✔
✔	10 2 2 2	191	191	✔

	4 4 3 3 5 5 5			
✔	2 45 3	45	45	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

5-G-Product of Array elements-Minimum

Started on	Tuesday, 2 September 2025, 3:06 PM
State	Finished
Completed on	Tuesday, 2 September 2025, 3:06 PM
Time taken	11 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 two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.

For example:

Input	Result
3 1 2 3 4 5	28

6

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<stdlib.h>
int comp1(const void *a,const void *b){
    return (*(int *)a-*(int *)b);
}

int comp2(const void *a,const void *b){
    return (*(int *)b-*(int *)a);
}

int main(){
    int n;
    scanf("%d",&n);
    int arr1[n],arr2[n];
    for(int i=0;i<n;i++) scanf("%d",&arr1[i]);
    for(int i=0;i<n;i++) scanf("%d",&arr2[i]);
    qsort(arr1,n,sizeof(int),comp1);
    qsort(arr2,n,sizeof(int),comp2);
    int pro=0;
    for(int i=0;i<n;i++){
        pro+=arr1[i]*arr2[i];
    }
    printf("%d",pro);
}
```

	Input	Expected	Got	
✔	3 1 2 3 4 5 6	28	28	✔
✔	4 7 5 1 2 1	22	22	✔

	3 4 1			
✔	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.

Finish review

Back to Course

1-DP-Playing with Numbers

Started on	Monday, 3 November 2025, 6:27 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:27 PM
Time taken	18 secs
Grade	**10.00** out of 10.00 (**100**%)

Question 1 | Correct Mark 10.00 out of 10.00 Flag question

Playing with Numbers:

Ram and Sita are playing with numbers by giving puzzles to each other. Now it was Ram term, so he gave Sita a positive integer 'n' and two numbers 1 and 3. He asked her to find the possible ways by which the number n can be represented using 1 and 3.Write any efficient algorithm to find the possible ways.

Example 1:

Input: *6*
Output:*6*
Explanation: *There are 6 ways to 6 represent number with 1 and 3*

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

First Line contains the number n

Output Format

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

Sample Input

6

Sample Output

6

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <stdint.h>

int main() {
    int n;
    if (scanf("%d", &n) != 1) return 0;

    if (n < 0) {
        printf("0\n");
        return 0;
    }

    long long dp[n+1];
    for (int i = 0; i <= n; ++i) dp[i] = 0;

    dp[0] = 1;
    for (int i = 1; i <= n; ++i) {
        dp[i] = dp[i-1];
        if (i >= 3) dp[i] += dp[i-3];
    }

    printf("%llu\n", dp[n]);
    return 0;
}
```

	Input	Expected	Got	

	Input	Expected	Got	
✔	6	6	6	✔
✔	25	8641	8641	✔
✔	100	24382819596721629	24382819596721629	✔

Passed all tests! ✔

Correct

Marks for this submission: 10.00/10.00.

Finish review

Back to Course

2-DP-Playing with chessboard

Started on	Monday, 3 November 2025, 6:28 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:28 PM
Time taken	25 secs
Grade	**10.00** out of 10.00 (**100**%)

Question 1 Correct Mark 10.00 out of 10.00 Flag question

Playing with Chessboard:

Ram is given with an n*n chessboard with each cell with a monetary value. Ram stands at the (0,0), that the position of the top left white rook. He is been 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

1 2 4

2 3 4

8 7 1

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

The next n lines contain the n*n chessboard values

Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    int n;
    scanf("%d", &n);

    int board[n][n];
    int dp[n][n];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &board[i][j]);
        }
    }

    dp[0][0] = board[0][0];

    for (int j = 1; j < n; j++) {
        dp[0][j] = dp[0][j - 1] + board[0][j];
    }

    for (int i = 1; i < n; i++) {
        dp[i][0] = dp[i - 1][0] + board[i][0];
    }

    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            dp[i][j] = board[i][j] + max(dp[i - 1][j], dp[i][j - 1]);
```

```
    }
}

printf("%d\n", dp[n - 1][n - 1]);

return 0;
}
```

	Input	Expected	Got	
✔	3 1 2 4 2 3 4 8 7 1	19	19	✔
✔	3 1 3 1 1 5 1 4 2 1	12	12	✔
✔	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✔

Passed all tests! ✔

Correct

Marks for this submission: 10.00/10.00.

Finish review

Back to Course

Data retention summary

3-DP-Longest Common Subsequence

Started on	Monday, 3 November 2025, 6:28 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:28 PM
Time taken	16 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 two strings find the length of the common longest subsequence(need not be contiguous) between the two.

Example:

s1: ggtabe

s2: tgatasb

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

The length is 4

Solveing it using Dynamic Programming

For example:

Input	Result
aab azb	2

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <string.h>

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    char s1[1000], s2[1000];
    scanf("%s", s1);
    scanf("%s", s2);

    int n = strlen(s1);
    int m = strlen(s2);

    int dp[n + 1][m + 1];

    for (int i = 0; i <= n; i++) {
        for (int j = 0; j <= m; j++) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (s1[i - 1] == s2[j - 1])
                dp[i][j] = 1 + dp[i - 1][j - 1];
            else
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }

    printf("%d\n", dp[n][m]);

    return 0;
}
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

4-DP-Longest non-decreasing Subsequence

Started on	Monday, 3 November 2025, 6:29 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:29 PM
Time taken	17 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:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Eg:

Input:9

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

the subsequence is [-1,2,2,2,2,3]

Output:6

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int dp[n];

    int max_len = 1;

    for (int i = 0; i < n; i++) {
        dp[i] = 1;
        for (int j = 0; j < i; j++) {
            if (arr[j] <= arr[i]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
        if (dp[i] > max_len)
            max_len = dp[i];
    }

    printf("%d\n", max_len);

    return 0;
}
```

	Input	Expected	Got	
✔	9 -1 3 4 5 2 2 2 2 3	6	6	✔
✔	7 1 2 2 4 5 7 6	6	6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity

Started on	Monday, 3 November 2025, 6:29 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:29 PM
Time taken	15 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

Find Duplicate in Array.

Given a read only 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 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int duplicate = -1;

    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                duplicate = arr[i];
                break;
            }
        }
        if (duplicate != -1)
            break;
    }

    printf("%d\n", duplicate);

    return 0;
}
```

	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.

Finish review

Back to Course

2-Finding Duplicates-O(n) Time Complexity,O(1) Space Complexity

Started on	Monday, 3 November 2025, 6:29 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:30 PM
Time taken	15 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question 1 Correct Mark 1.00 out of 1.00 Flag question

Find Duplicate in Array.

Given a read only 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 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int slow = arr[0];
    int fast = arr[0];

    do {
        slow = arr[slow];
        fast = arr[arr[fast]];
    } while (slow != fast);

    slow = arr[0];
    while (slow != fast) {
        slow = arr[slow];
        fast = arr[fast];
    }

    printf("%d\n", slow);
    return 0;
}
```

	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.

Finish review

Back to Course

3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Space Complexity

Started on	Monday, 3 November 2025, 6:30 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:30 PM
Time taken	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 Flag question

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 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        int n1, n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++)
            scanf("%d", &arr1[i]);

```

```
        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++)
            scanf("%d", &arr2[i]);

        for (int i = 0; i < n1; i++) {
            for (int j = 0; j < n2; j++) {
                if (arr1[i] == arr2[j]) {
                    printf("%d ", arr1[i]);
                    break;
                }
            }
        }
        printf("\n");
    }

    return 0;
}
```

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary

4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity

Started on	Monday, 3 November 2025, 6:30 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:30 PM
Time taken	17 secs
Marks	1.00/1.00
Grade	**30.00** out of 30.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        int n1, n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++)
            scanf("%d", &arr1[i]);

```

```c
        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++)
            scanf("%d", &arr2[i]);

        int i = 0, j = 0;
        while (i < n1 && j < n2) {
            if (arr1[i] == arr2[j]) {
                printf("%d ", arr1[i]);
                i++;
                j++;
            }
            else if (arr1[i] < arr2[j]) {
                i++;
            }
            else {
                j++;
            }
        }
        printf("\n");
    }

    return 0;
}
```

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary

5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Complexity

Started on	Monday, 3 November 2025, 6:31 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:31 PM
Time taken	15 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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 != j.

Input Format:

First Line n - 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 1 3 5 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n, k;
    scanf("%d", &n);

    int A[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &A[i]);

    scanf("%d", &k);

    int found = 0;

    for (int i = 0; i < n - 1 && !found; i++) {
        for (int j = i + 1; j < n; j++) {
            int diff = A[j] - A[i];
            if (diff == k) {
                found = 1;
                break;
            } else if (diff > k) {
                break;
            }
        }
    }

    printf("%d", found);

    return 0;
}
```

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

6-Pair with Difference -O(n) Time Complexity,O(1) Space Complexity

Started on	Monday, 3 November 2025, 6:31 PM
State	Finished
Completed on	Monday, 3 November 2025, 6:31 PM
Time taken	19 secs
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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 != j.

Input Format:

First Line n - 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 1 3 5 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n, k;
    scanf("%d", &n);

    int A[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &A[i]);

    scanf("%d", &k);

    int i = 0, j = 1;
    int found = 0;

    while (i < n && j < n) {
        if (i != j) {
            int diff = A[j] - A[i];

            if (diff == k) {
                found = 1;
                break;
            } else if (diff < k) {
                j++;
            } else {
                i++;
            }
        } else {
            j++;
        }
    }

```

```
33      printf("%d", found);
34
35      return 0;
36  }
37
```

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course