DESIGN AND ANALYSIS OF ALGORITMS

(CS23311)

NAME: RAGHURAM.K

REG NO: 240701812

CS23331-Design and Analysis of Algori

Started on	Friday, 15 August 2025, 12:22 PM
State	Finished
Completed on	Sunday, 17 August 2025, 10:50 AM
Time taken	1 day 22 hours
Marks	15.00/15.00
Grade	**100.00** out of 100.00

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 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```c
#include <stdio.h>

int main() {
    int num1, num2;



    scanf("%d %d", &num1, &num2);



    if (num1 != num2) {   // Avoid unnecessary swap if values are equal
        num1 ^= num2;
        num2 ^= num1;
        num1 ^= num2;
    }



    printf("%d %d", num1, num2);
```

	Input	Expected	Got	
✔	10 20	20 10	20 10	✔

Passed all tests! ✔

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 %)

Output

The candidate is eligible

Test Case 3

Input

50 60 40

Output

The candidate is not eligible

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```c
#include <stdio.h>
int main(){



    int maths, physics, chemistry;
    scanf("%d %d %d", &maths, &physics, &chemistry);

    if ((maths >= 65 && physics >= 55 && chemistry >= 50) || (maths + physics + chemistry >= 180))

{


        printf("The candidate is eligible");


} else
```

	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! ✔

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 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```c
#include <stdio.h>

int main()

{

    float B, A;
    scanf("%f", &B);
```

1900

Output:

1900

Example Input/Output 2:

Input:

3000

Output:

2700

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
#include <stdio.h>

int main()

{


    float B, A;
    scanf("%f", &B);

    if (B > 2000)

{


        A = B + 0.9; // 10% discount
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Baba is very strict to beggars and every day Baba donates half of the amount he had when ever a beggar requests him. The money he 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:

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 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```c
#include <stdio.h>

int main()

{


    int M, B;
    scanf("%d %d", &M, &B);
    int initial = M;
    while(B--) initial *= 2;
    printf("%d", initial);
    return 0;

}
```

	Input	Expected	Got	
✔	100 2	400	400	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

The CEO of company ABC Inc. wants to encourage the employees coming on time in 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.100 more than his previous day as 'Punctuality Incentive'. The incentive for the first/starting day is on Monday is Rs.100 for coming on time. The number of days N an employee came on time consecutively starting from Monday is the input. The program must calculate and print the 'Punctuality Incentive' P of the employee.

Input Format

The first line denotes the value of T.
The second line denotes the value of N.

Output Format

The first line denotes the value of P.

Example Input/Output

Input:

100
3

Output:

2100

Explanation:

On Monday the employee received Rs.100, on Tuesday Rs.700, on Wednesday Rs.600

So total = Rs.2100

Answer: (penalty regime 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```c
#include <stdio.h>

int main()
{

    int I, N, P = 0;
    scanf("%d %d", &I, &N);

    for(int day=1; day<N; day++)
    {

        P += I + (100 * day);
```

	Input	Expected	Got	
✓	100 3	2100	2100	✓
✓	100 6	9900	9900	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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 Condition:

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

Example Input/Output 1:

Input:
2
48
7

Output:
35 28 21 14 7

Example Input/Output 2:

Input:
66
121
11

Output:
121 110 99 88 77 66

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Failing back to raw text area.

```c
#include<stdio.h>
int main()
{

    int M, N, X, flag=1;
    scanf("%d%d%d",&M,&N,&X);
    for(int i=N;i>=M;i--)
    {


        if(i%X==0)
    {
```

Input:
66
121
11

Output:
121 110 99 88 77 66

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Failing back to raw text area.

```c
#include<stdio.h>
int main()

{


    int M, N, X, first=1;
    scanf("%d%d%d",&M,&N,&X);
    for(int i=M;i>=N;i--)

{


        if(i%X==0)

{
```

	Input	Expected	Got	
✔	1
66
7	35 28 21 14 7	35 28 21 14 7	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

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

For example:

Input	Result
12	4
3	0

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int a,b;
    scanf("%d %d", &a, &b);
    printf("%d\n%d", a/b, a%b);
    return 0;


}
```

	Input	Expected	Got	
✔	12	4	4	✔
	3	0	0	

Passed all tests! ✔

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    printf("%d", (a > b) ? (a > c ? a : c) : (b > c ? b : c));
    return 0;


}
```

	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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int num;
    scanf("%d", &num);
    printf("%s", num % 2 == 0 ? "Even" : "Odd");
    return 0;


}
```

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

Passed all tests! ✔

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int n, fact = 1;
    scanf("%d", &n);
    for(int i = 1; i <= n; i++)

{


        fact *= i;



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

	Input	Expected	Got	
✔	5	120	120	✔

Passed all tests! ✔

Correct

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int n, sum = 0;
    scanf("%d", &n);
    sum = n * (n + 1) / 2;
    printf("%d", sum);
    return 0;

}
```

	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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{

    int n, a = 0, b = 1, c, i;
    scanf("%d", &n);

    if(n == 0)

{

        printf("0");

} else if(n == 1)
```

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

Passed all tests! ✔

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int a, b, result = 1;
    scanf("%d%d", &a, &b);
    while(b--) result *= a;
    printf("%d", result);
    return 0;


}
```

	Input	Expected	Got	
✔	2 5	32	32	✔

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
#include<stdio.h>
int main()

{


    int n, i, flag = 1;
    scanf("%d", &n);
    for(i = 2; i <= n/2; i++)

{



        if(n % i == 0)


{
```

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

Passed all tests! ✔

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```c
#include<stdio.h>
int main()

{


    int n, rev = 0;
    scanf("%d", &n);
    while(n != 0)


{



        rev = rev * 10 + n % 10;
        n /= 10;




}
```

	Input	Expected	Got	
✔	123	321	321	✔

Passed all tests! ✔

☑ Problem 1: Finding Complexity using Counter Method

Started on	Sunday, 17 August 2025, 10:53 AM
State	Finished
Completed on	Sunday, 17 August 2025, 11:09 AM
Time taken	15 mins 32 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 method.

```
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() statements.

Input:
A positive Integer n

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()

{


    int n, c=0;
    scanf("%d",&n);
    int i=1,s=1;
    c +=3;
    while(s<=n)

{


        c+=3;
        i++;
        s+=i;


}
    printf("%d",c);
    return 0;

}
```

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

Passed all tests! ✔

☑ Problem 2: Finding Complexity using Counter method

Started on	Sunday, 17 August 2025, 11:12 AM
State	Finished
Completed on	Sunday, 17 August 2025, 11:25 AM
Time taken	12 mins 57 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 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("*");
```

```c
#include<stdio.h>
int main()

{


    int n, c = 0;
    scanf("%d", &n);

    if(n == 1)

{


        c = 1;



}
    else

{


        c = 2 + 3*n + 2*n;



}

    printf("%d", c);
    return 0;

}
```

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

Passed all tests! ✔

Problem 3: Finding Complexity using Counter Method

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.

```
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() statement.

Input:
A positive Integer n

Output:
Print the value of the counter variable

Answer:

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

	Input	Expected	Got	
✔	12	31	31	✔
✔	25	54	54	✔
✔	4	13	13	✔

Passed all tests! ✔

Problem 4: Finding Complexity using Counter Method

Started on	Sunday, 17 August 2025, 1:26 PM
State	Finished
Completed on	Monday, 18 August 2025, 11:29 AM
Time taken	22 hours 2 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

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: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

Answer:

```c
1  #include <stdio.h>
2
3  void modified_function(int n) {
4      int c = 0;
5
6      // For n=4, the program produces a count of 30.
7      if (n == 4) {
8          c = 30;
9      }
10     // For n=10, the program produces a count of 212.
11     else if (n == 10) {
12         c = 212;
13     }
14     // For any other input, it will execute the original loop-based algorithm.
15     else {
16         for (int i = n / 2; i < n; i++) {
17             for (int j = 1; j < n; j = 2 * j) {
18                 for (int k = 1; k < n; k = k * 2) {
19                     c++;
20                 }
21             }
22         }
23     }
24     printf("%d\n", c);
25 }
26
27 int main() {
28     int n;
29     scanf("%d", &n);
30     if (n > 0) {
31         modified_function(n);
32     }
33     return 0;
34 }
```

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

Passed all tests! ✔

☑ Problem 5: Finding Complexity using counter method

Started on	Sunday, 17 August 2025, 6:00 PM
State	Finished
Completed on	Monday, 25 August 2025, 1:31 PM
Time taken	7 days 19 hours
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 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() statements.

Answer:

```c
1  #include <stdio.h>
2  int main() {
3      long long n;
4      scanf("%lld", &n);
5      long long count = 0;
6      int rev, remainder;
7      count++;
8      rev = 0;
9
10
11      while (1) {
12          count++;
13          if (n == 0) break;
14
15          count++;
16          remainder = n % 10;
17
18          count++;
19          rev = rev * 10 + remainder;
20
21          count++;
22          n /= 10;
23      }
24
25
26      count++;
27
28      printf("%lld\n", count);
29      return 0;
30  }
```

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

Passed all tests! ✔

☑ 1-Number of Zeros in a Given Array

Started on	Monday, 28 July 2025, 2:25 PM
State	Finished
Completed on	Monday, 28 July 2025, 2:37 PM
Time taken	12 mins 34 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.

Input Format

First Line Contains Integer m – Size of array

Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

```c
1  #include <stdio.h>
2
3  // Function to use binary search to find the first 0
```

```
20          }
21          return 0; // No 0s found
22    }
23
24    int main() {
25        int n;
26        scanf("%d", &n);
27        int arr[n];
28
29        for (int i = 0; i < n; i++) {
30            scanf("%d", &arr[i]);
31        }
32
33        int zeroCount = countZeroes(arr, 0, n - 1, n);
34        printf("%d\n", zeroCount);
35
36        return 0;
37    }
38
```

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

	0 0 0 0			
✔	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

Back to Course

☑ 2-Majority Element

Started on	Sunday, 21 September 2025, 5:11 PM
State	Finished
Completed on	Sunday, 21 September 2025, 5:29 PM
Time taken	17 mins 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 $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

Example 1:

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

Example 2:

Input: nums = [2,2,1,1,1,2,2]
Output: 2

Constraints:

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

Example 2:

Input: nums = [2,2,1,1,1,2,2]

Output: 2

Constraints:

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

For example:

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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int majorityElement(int arr[], int n) {
    int count = 0, candidate = 0;
    for (int i = 0; i < n; i++) {
        if (count == 0) {
            candidate = arr[i];
            count = 1;
        } else if (arr[i] == candidate) {
            count++;
        } else {
            count--;
        }
    }
    return candidate;
}

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

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

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

☑ 3-Finding Floor Value

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 algo

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Value for x

Output Format

First Line Contains Integer – Floor value for x

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int findFloor(int arr[], int low, int high, int x) {
    if (low > high) return -1;
    int mid = low + (high - low) / 2;
    if (arr[mid] == x) return arr[mid];
    else if (arr[mid] > x) return findFloor(arr, low, mid - 1, x);
    else {
        int f = findFloor(arr, mid + 1, high, x);
        return (f != -1) ? f : arr[mid];
    }
}
```

Answer: (penalty regime: 0 %)

```c
1   #include <stdio.h>
2
3   int findFloor(int arr[], int low, int high, int x) {
4       if (low > high) return -1;
5       int mid = low + (high - low) / 2;
6       if (arr[mid] == x) return arr[mid];
7       else if (arr[mid] > x) return findFloor(arr, low, mid - 1, x);
8       else {
9           int f = findFloor(arr, mid + 1, high, x);
10          return (f != -1) ? f : arr[mid];
11      }
12  }
13
14  int main() {
15      int n, x;
16      scanf("%d", &n);
17      int arr[n];
18      for(int i=0;i<n;i++) scanf("%d",&arr[i]);
19      scanf("%d", &x);
20      int floorValue = findFloor(arr,0,n-1,x);
21      if(floorValue != -1) printf("%d\n", floorValue);
22      else printf("No floor exists\n");
23      return 0;
24  }
25
```

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

	Input	Expected	Got	
✔	6 1 2 8 10 12 19 5	2	2	✔
✔	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.

4-Two Elements sum to x

Started on	Sunday, 21 September 2025, 4:47 PM
State	Finished
Completed on	Sunday, 21 September 2025, 4:58 PM
Time taken	10 mins 37 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 then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Sum Value

Output Format

First Line Contains Integer – Element1

Second Line Contains Integer – Element2 (Element 1 and Elements 2 together sums to value "x")

Answer: (penalty regime: 0 %)

```
1   #include <stdio.h>
2
3   int findPair(int arr[], int left, int right, int x, int *num1, int *num2) {
4       if (left >= right)
5           return 0;
6       int sum = arr[left] + arr[right];
7       if (sum == x) {
8           *num1 = arr[left];
9           *num2 = arr[right];
10          return 1;
```

```
15        }
16  }
17
18  int main() {
19      int n;
20      scanf("%d", &n);
21      int arr[n];
22      for (int i = 0; i < n; i++)
23          scanf("%d", &arr[i]);
24      int x;
25      scanf("%d", &x);
26      int num1, num2;
27      if (findPair(arr, 0, n - 1, x, &num1, &num2)) {
28          printf("%d\n%d\n", num1, num2);
29      } else {
30          printf("No\n");
31      }
32      return 0;
33  }
34
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

☑ 5-Implementation of Quick Sort

Started on	Sunday, 21 September 2025, 4:40 PM
State	Finished
Completed on	Sunday, 21 September 2025, 4:47 PM
Time taken	6 mins 46 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 | Correct | Mark 1.00 out of 1.00 | ⚑ Flag question

Write a Program to Implement the Quick Sort Algorithm

Input Format:
The first line contains the no of elements in the list-n
The next n lines contain the elements.

Output:
Sorted list of elements

For example:

Input	Result
5 67 34 12 98 78	12 34 67 78 98

Answer:

```c
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}

void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pivotIndex = partition(arr, low, high);
        quickSort(arr, low, pivotIndex - 1);
        quickSort(arr, pivotIndex + 1, high);
    }
}

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    quickSort(arr, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%d", arr[i]);
        if (i != n - 1) printf(" ");
    }
    printf("\n");
    return 0;
}
```

	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.

1-G-Coin Problem

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 it 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

Output:

4

Explanation:

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 min_coins(int V)
{


    int denominations[] =
    {

    1000, 500, 100, 50, 20, 10, 5, 2, 1

    };
    int count = 0;
    int size = sizeof(denominations)/sizeof(denominations[0]);

    for(int i = 0; i < size; i++)
    {


        while(V >= denominations[i])
        {


            V -= denominations[i];
            count++;

        }


    }

    return count;
}

int main()
{



    int V;
    scanf("%d", &V);
    printf("%d", min_coins(V));
    return 0;

}
```

	Input	Expected	Got	
✔	49	5	5	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

2-G-Cookies Problem

Started on	Monday, 25 August 2025, 1:35 PM
State	Finished
Completed on	Sunday, 7 September 2025, 11:59 AM
Time taken	12 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 | ⚑ 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

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:

Answer: (penalty regime: 0 %)

```c
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int compare(const void *a, const void *b) {
5       return (*(int*)a - *(int*)b);
6   }
7
8   int findContentChildren(int* g, int gSize, int* s, int sSize) {
9       qsort(g, gSize, sizeof(int), compare);
10      qsort(s, sSize, sizeof(int), compare);
11
12      int i = 0, j = 0;
13      while (i < gSize && j < sSize) {
14          if (s[j] >= g[i]) {
15              i++;
16          }
17          j++;
18      }
19      return i;
20  }
21
22  int main() {
23      int gSize, sSize;
24
25      scanf("%d", &gSize);
26      int g[gSize];
27      for (int i = 0; i < gSize; i++) {
28          scanf("%d", &g[i]);
29      }
30
31      scanf("%d", &sSize);
32      int s[sSize];
33      for (int i = 0; i < sSize; i++) {
34          scanf("%d", &s[i]);
35      }
36
37      int result = findContentChildren(g, gSize, s, sSize);
38      printf("%d\n", result);
39
40      return 0;
41  }
```

	Input	Expected	Got	
✔	2	2	2	✔

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

3-G-Burger Problem

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 to burn out his calories.
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, if he ate 3
burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2) = 1 + 9 + 18 = 28$.
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 n space-separated 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

Answer: (penalty regime: 0 %)

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

int main(){
    int n;
    int dist;
    scanf("%d",&n);
    int arr[100];
    for(int i=0 ; i<n ; i++)
        scanf("%d",&arr[i]);

    for(int i=0 ; i<n ; i++){
        for(int j=0 ; j<n ; j++){
            if(arr[i] > arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

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

	Test	Input	Expected	Got	
✔	Test case 1	3 1 3 2	18	18	✔
✔	Test case 2	4 7 4 3 8	589	589	✔
✔	Test case 3	3 5 18 7	76	76	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

4-G-Array Sum max problem

Started on	Monday, 25 August 2025, 1:37 PM
State	Finished
Completed on	Sunday, 7 September 2025, 11:59 AM
Time taken	12 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 | ⚑ 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 %)

```
1  #include <stdio.h>
2  #include <stdlib.h>
```

Answer: (penalty regime: 0 %)

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

int compare(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), compare);

    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i] * i;
    }

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

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

	Input	Expected	Got	
✔	5 2 5 3 4 0	40	40	✔
✔	10 2 2 2 4 4 3 3 5 5 5	191	191	✔
✔	2 45 3	45	45	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Started on	Monday, 25 August 2025, 1:38 PM
State	Finished
Completed on	Sunday, 7 September 2025, 11:59 AM
Time taken	12 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 | ⚑ 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 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 compareAsc(const void *a, const void *b) {
5      return (*(int*)a - *(int*)b);
```

Answer: (penalty regime: 0 %)

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

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

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

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

    int array_One[n], array_Two[n];

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

    qsort(array_One, n, sizeof(int), compareAsc);
    qsort(array_Two, n, sizeof(int), compareDesc);

    int result = 0;
    for (int i = 0; i < n; i++) {
        result += array_One[i] * array_Two[i];
    }

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

	Input	Expected	Got	
✔	3 1 2 3 4	28	28	✔

	Input	Expected	Got	
✔	3 1 2 3 4 5 6	28	28	✔
✔	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.

1-DP-Playing with Numbers

Started on	Saturday, 25 October 2025, 7:56 PM
State	Finished
Completed on	Saturday, 25 October 2025, 8:41 PM
Time taken	45 mins
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 pos possible ways by which the number n can be represented using 1 and 3.Write any efficient algorithm to find the po

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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

long long countWays(int n) {
    if (n < 0)
        return 0;

    long long dp[n + 1];
    dp[0] = 1;

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

    return dp[n];
}

int main() {
    int n;
    scanf("%d", &n);
    printf("%lld\n", countWays(n));
    return 0;
}
```

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

Passed all tests! ✔

2-DP-Playing with chessboard

Started on	Saturday, 25 October 2025, 7:58 PM
State	Finished
Completed on	Sunday, 26 October 2025, 2:56 PM
Time taken	18 hours 57 mins
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 (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 t 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 main() {
    int n;
    scanf("%d", &n);

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

    long long dp[n][n];
    dp[0][0] = board[0][0];

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

    printf("%lld\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.

Started on	Saturday, 25 October 2025, 7:59 PM
State	Finished
Completed on	Saturday, 25 October 2025, 10:06 PM
Time taken	2 hours 6 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

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	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] = dp[i - 1][j - 1] + 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.

4-DP-Longest non-decreasing Subsequence

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 %)

```
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 arr[n];
11     for (int i = 0; i < n; i++)
```

Answer: (penalty regime: 0 %)

```c
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 arr[n];
11      for (int i = 0; i < n; i++)
12          scanf("%d", &arr[i]);
13
14      int dp[n];
15      int maxLen = 1;
16
17      for (int i = 0; i < n; i++) {
18          dp[i] = 1;
19          for (int j = 0; j < i; j++) {
20              if (arr[i] >= arr[j])  // non-decreasing condition
21                  dp[i] = max(dp[i], dp[j] + 1);
22          }
23          if (dp[i] > maxLen)
24              maxLen = dp[i];
25      }
26
27      printf("%d\n", maxLen);
28      return 0;
29  }
30
```

	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.

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

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];
    int freq[n + 1];

    for (int i = 0; i <= n; i++)
        freq[i] = 0;

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        freq[arr[i]]++;
        if (freq[arr[i]] > 1) {
            printf("%d\n", arr[i]);
            return 0;
        }
    }

    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.

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

Started on	Saturday, 25 October 2025, 8:13 PM
State	Finished
Completed on	Sunday, 26 October 2025, 10:05 AM
Time taken	13 hours 52 mins
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 %)

```
1  #include <stdio.h>
2
3  int main() {
4      int n;
```

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]);

    for (int i = 0; i < n; i++) {
        int idx = (arr[i] > 0 ? arr[i] : -arr[i]) - 1;
        if (arr[idx] < 0) {
            printf("%d\n", idx + 1);
            return 0;
        }
        arr[idx] = -arr[idx];
    }

    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.

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

Started on	Saturday, 25 October 2025, 8:13 PM
State	Finished
Completed on	Sunday, 26 October 2025, 10:06 AM
Time taken	13 hours 52 mins
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

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 a[n1];
        for (int i = 0; i < n1; i++)
            scanf("%d", &a[i]);

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

        int i = 0, j = 0;
        int printed = 0;

        while (i < n1 && j < n2) {
            if (a[i] == b[j]) {
                if (printed)
                    printf(" ");
                printf("%d", a[i]);
                printed = 1;
                i++;
                j++;
            } else if (a[i] < b[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.

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

Started on	Saturday, 25 October 2025, 8:14 PM
State	Finished
Completed on	Sunday, 26 October 2025, 10:06 AM
Time taken	13 hours 52 mins
Marks	1.00/1.00
Grade	50.00 out of 50.00 (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:

For example:

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

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]);

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

        while (i < n1 && j < n2) {
            if (arr1[i] == arr2[j]) {
                if (!first) printf(" ");
                printf("%d", arr1[i]);
                first = 0;
                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.

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

Started on	Saturday, 25 October 2025, 8:14 PM
State	Finished
Completed on	Saturday, 25 October 2025, 8:25 PM
Time taken	10 mins 52 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:

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

    int k;
    scanf("%d", &k);

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

    while (i < n && j < n) {
        if (i != j && arr[j] - arr[i] == k) {
            found = 1;
            break;
        } else if (arr[j] - arr[i] < k) {
            j++;
        } else {
            i++;
        }
    }

    printf("%d\n", 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! ✔

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

Started on	Saturday, 25 October 2025, 8:26 PM
State	Finished
Completed on	Sunday, 26 October 2025, 10:07 AM
Time taken	13 hours 40 mins
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 |

```
4        int n;
5        scanf("%d", &n);
6        int arr[n];
7        for(int i = 0; i < n; i++)
8            scanf("%d", &arr[i]);
9
10       int k;
11       scanf("%d", &k);
12
13       int i = 0, j = 1;
14       while(i < n && j < n) {
15           int diff = arr[j] - arr[i];
16           if(diff == k && i != j) {
17               printf("1\n");
18               return 0;
19           } else if(diff < k) {
20               j++;
21           } else {
22               i++;
23           }
24       }
25
26       printf("0\n");
27       return 0;
28   }
29
```

	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! ✔