

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

Started on Wednesday, 24 September 2025, 3:09 PM

State Finished

Completed on Wednesday, 8 October 2025, 4:09 PM

Time taken 14 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

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,temp;
5     scanf("%d %d",&a,&b);
6     temp = a;
7     a = b;
8     b = temp;
9     printf("%d %d",a,b);
10 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

Correct

---

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

Marks in Maths  $\geq$  65

Marks in Physics  $\geq$  55

Marks in Chemistry  $\geq$  50

Or

Total in all three subjects  $\geq$  180

### **Sample Test Cases**

#### **Test Case 1**

##### **Input**

70 60 80

##### **Output**

The candidate is eligible

#### **Test Case 2**

## **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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c,total;
5     scanf("%d %d %d",&a,&b,&c);
6     total = a+b+c;
7     if((a>=65 && b>=55 && c>=50) || (total>=180))
8         printf("The candidate is eligible");
9     else
10        printf("The candidate is not eligible");
11
12 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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:

## Example Input/Output 2:

Input:

3000

Output:

2700

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

```
1 #include<stdio.h>
2 int main()
3 {
4     int bill,final_bill,dis;
5     scanf("%d",&bill);
6     if(bill>2000){
7         dis = bill*0.10;
8         final_bill = bill-dis;
9         printf("%d",final_bill);
10    }
11 }
12 else{
13     printf("%d",bill);
14 }
15 return 0;
16 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 whenever 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:**

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

**Example Input/Output:**

Input:

```
100  
2
```

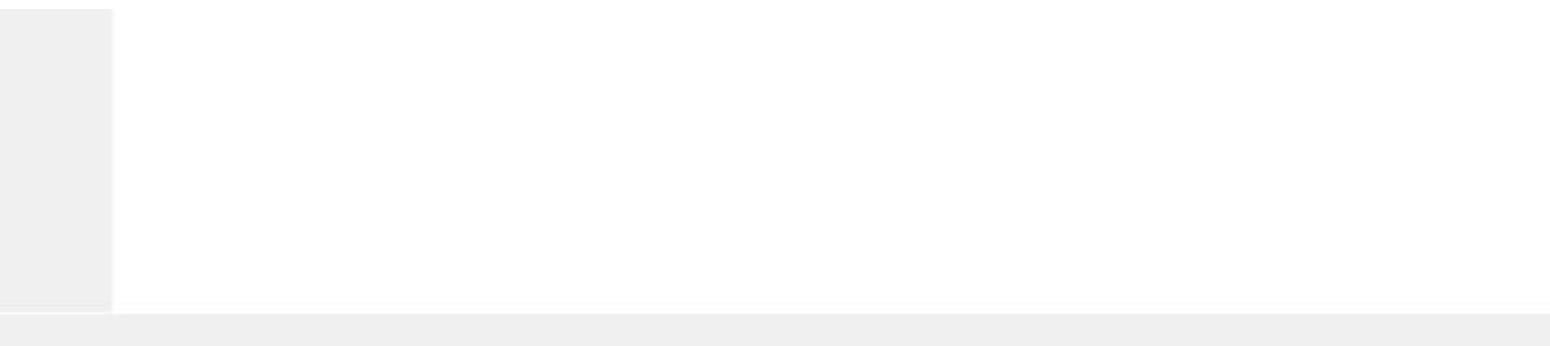
Output:

Explanation:

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

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

```
1 #include<stdio.h>
2 int main(){
3     int bal;
4     int n;
5     scanf("%d%d",&bal,&n);
6     for(int i=0;i<n;i++)
7     {
8         bal = 2*bal;
9     }
10    printf("%d",bal);
11    return 0;
12 }
```



	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int cost;
5     int n;
6     scanf("%d%d",&cost,&n);
7     int prev=cost;
8     for(int i=0;i<n-1;i++){
9         prev=prev+200;
10        cost=cost+prev;
11    }
12    printf("%d",cost);
13    return 0;
14 }
```



	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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:**

$1 \leq M \leq 9999999$   
 $M < N \leq 9999999$   
 $1 \leq X \leq 9999$

**Example Input/Output 1:**

Input:  
2  
40  
7

Output:  
35 28 21 14 7

**Example Input/Output 2:**

## Example Input/Output 2:

Input:

66  
121  
11

Output:

121 110 99 88 77 66

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,m,x;
5     scanf("%d %d %d",&n,&m,&x);
6     for (int i=m;i>n-1;i--){
7         if(i%x==0)
8             {
9                 printf("%d ",i);
10            }
11        }
12    return 0;
13 }
```

	Input	Expected	Got	
✓	2	35 28 21 14 7	35 28 21 14 7	✓
	40			
	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 remainder of given integers.

**For example:**

Input	Result
12	4
3	0

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

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     scanf("%d %d",&a,&b);
6     int Q,R;
7     Q = a/b;
8     R = a%b;
9     printf("%d \n%d",Q,R);
10    return 0;
11 }
```

	Input	Expected	Got	
✓	12	4	4	✓
	3	0	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     scanf("%d %d %d",&a,&b,&c);
6     if(a>b && a>c)
7         printf("%d",a);
8     else if(b>a && b>c)
9         printf("%d",b);
10    else
11        printf("%d",c);
12 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a;
5     scanf("%d",&a);
6     if(a %2 == 0)
7         printf("Even");
8     else
9         printf("Odd");
10 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,i;
5     long long fact = 1;
6     scanf("%d",&n);
7     for(i=1;i<=n;i++){
8         fact = fact*i;
9     }
10    printf("%lld",fact);
11    return 0;
12 }
```

	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,sum=0,i;
5     scanf("%d",&n);
6     for(i=1;i<=n;i++){
7         sum += i;
8     }
9     printf("%d",sum);
10    return 0;
11 }
```

	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     if(n==0) {printf("0"); return 0;}
7     if(n==1) {printf("1"); return 0;}
8     long long a=0, b=1, c;
9     for(int i=2;i<=n;i++) {
10         c=a+b;
11         a=b;
12         b=c;
13     }
14     printf("%lld",b);
15     return 0;
16 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,power=1;
5     scanf("%d %d",&a,&b);
6     while(b>0){
7         power=power*a;
8         b--;
9     }
10    printf("%d",power);
11    return 0;
12 }
```

	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int num,i,count=0;
5     scanf("%d",&num);
6     for(i=1;i<=num;i++){
7         if(num%i==0)
8             count++;
9     }
10    if(count==2)
11        printf("Prime");
12    else
13        printf("No Prime");
14    return 0;
15 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int num,rev=0;
5     scanf("%d",&num);
6     while(num != 0){
7         rev = rev*10 + num%10;
8         num = num/10;
9     }
10    printf("%d",rev);
11    return 0;
12 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	123	321	321	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.



 BINESH KARTHIK S 2024-CSE ▾

**B2**

**Started on** Tuesday, 12 August 2025, 8:34 AM

**State** Finished

**Completed on** Tuesday, 12 August 2025, 8:40 AM

**Time taken** 6 mins 30 secs

**Marks** 1.00/1.00

**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

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

**Output:**

Print the value of the counter variable

**For example:**

Input	Result
9	12

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

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

	Input	Expected	Got	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

[Back to Course](#)

 **BINESH KARTHIK S 2024-CSE** ▾**B2****Started on** Tuesday, 12 August 2025, 8:40 AM**State** Finished**Completed on** Tuesday, 12 August 2025, 8:44 AM**Time taken** 3 mins 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

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

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

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

**Input:**

A positive Integer n

**Output:**

Print the value of the counter variable

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

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     scanf("%d", &n);
5     int counter = 0;
6     if (n == 1) {
7     } else {
8         counter += n;
9         counter += n;
10        counter += (n + 1);
11        counter += 2 * n;
12        counter += 1;
13    }
14    printf("%d\n", counter);
15    return 0;
16 }
17 }
```

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

	Input	Expected	Got	
✓	143	717	717	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Tuesday, 12 August 2025, 8:44 AM**State** Finished**Completed on** Tuesday, 12 August 2025, 8:51 AM**Time taken** 6 mins 33 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

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

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

**Note:** 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 int main() {
3     int num, i;
4     int counter = 1;
5     scanf("%d", &num);
6     for (i = 1; i <= num; ++i) {
7         counter++; // for loop iteration
8         counter++; // if condition check
9         if (num % i == 0) {
10             counter++; // printf execution
11             // printf("%d ", i);
12         }
13     }
14     printf("%d\n", counter);
15     return 0;
16 }
17 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

 **BINESH KARTHIK S 2024-CSE** ▾**B2****Started on** Tuesday, 12 August 2025, 8:51 AM**State** Finished**Completed on** Tuesday, 12 August 2025, 8:56 AM**Time taken** 5 mins 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

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:**

A positive Integer n

**Output:**

Print the value of the counter variable

**Answer:**

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

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

[Back to Course](#)

 **BINESH KARTHIK S 2024-CSE** ▾**B2****Started on** Tuesday, 12 August 2025, 8:56 AM**State** Finished**Completed on** Tuesday, 12 August 2025, 9:02 AM**Time taken** 5 mins 59 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

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

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

    }
    print(rev);
}
```

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

**Input:**

A positive Integer n

**Output:**

Print the value of the counter variable

**Answer:**

```
1 #include <stdio.h>
2 int main() {
3     int n, rev = 0, remainder;
4     int counter = 1;
5     counter++;
6     scanf("%d", &n);
7     int temp = n;
8     while (temp != 0) {
9         counter++;
10        remainder = temp % 10;
11        counter++;
12        rev = rev * 10 + remainder;
13        counter++;
14        temp /= 10;
15        counter++;
16    }
17    counter++;
18    printf("%d\n", counter);
19    return 0;
20 }
21
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 24 September 2025, 8:43 PM**State** Finished**Completed on** Wednesday, 24 September 2025, 8:46 PM**Time taken** 3 mins 9 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00**Problem Statement**

Given 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.

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

```

1 #include <stdio.h>
2 int findFirstZero(int arr[], int low, int high) {
3     if (high >= low) {
4         int mid = low + (high - low) / 2;
5         if ((mid == 0 || arr[mid - 1] == 1) && arr[mid] == 0)
6             return mid;
7         if (arr[mid] == 1)
8             return findFirstZero(arr, mid + 1, high);
9         return findFirstZero(arr, low, mid - 1);
10    }
11    return -1;
12 }
13
14 int main() {
15     int m;
16     scanf("%d", &m);
17
18     int arr[m];
19
20     for (int i = 0; i < m; i++) {
21         scanf("%d", &arr[i]);
22     }
23
24     int firstZeroIndex = findFirstZero(arr, 0, m - 1);
25
26     int zeroCount = 0;
27     if (firstZeroIndex != -1) {
28         zeroCount = m - firstZeroIndex;
29     }
30
31     printf("%d\n", zeroCount);
32
33     return 0;
34 }
35

```

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 24 September 2025, 8:46 PM**State** Finished**Completed on** Wednesday, 24 September 2025, 8:51 PM**Time taken** 4 mins 55 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

Given 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 * 104`
- `-231 <= nums[i] <= 231 - 1`

**For example:**

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

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

```

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

35

	Input	Expected	Got	
✓	3 3 2 3	3	3 ✓	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 24 September 2025, 8:50 PM**State** Finished**Completed on** Wednesday, 24 September 2025, 8:54 PM**Time taken** 4 mins 6 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

**Problem Statement:**

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

**Input Format**

First Line Contains Integer n – Size of array  
 Next n lines Contains n numbers – Elements of an array  
 Last Line Contains Integer x – Value for x

**Output Format**

First Line Contains Integer – Floor value for x

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

```

1 #include <stdio.h>
2
3 int findFloor(int arr[], int low, int high, int x) {
4     int res = -1;
5     while (low <= high) {
6         int mid = (low + high) / 2;
7         if (arr[mid] == x)
8             return arr[mid];
9         else if (arr[mid] < x) {
10            res = arr[mid];
11            low = mid + 1;
12        } else {
13            high = mid - 1;
14        }
15    }
16    return res;
17 }
18
19 int main() {
20     int n;
21     scanf("%d", &n);
22
23     int arr[n];
24     for (int i = 0; i < n; i++)
25         scanf("%d", &arr[i]);
26
27     int x;
28     scanf("%d", &x);
29
30     int result = findFloor(arr, 0, n - 1, x);
31     printf("%d\n", result);
32
33     return 0;
34 }
```

	Input	Expected	Got	
✓	6	2	2	✓
	1			
	2			
	8			
	10			
	12			
	19			
	5			

	Input	Expected	Got	
✓	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 24 September 2025, 8:51 PM**State** Finished**Completed on** Wednesday, 24 September 2025, 8:54 PM**Time taken** 3 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**Problem Statement:**

Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution

**Input Format**

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Sum Value

**Output Format**

First Line Contains Integer – Element1

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

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

```

1 #include <stdio.h>
2
3 int arr[100000];
4
5 void findPair(int left, int right, int x) {
6     if (left >= right) {
7         printf("No\n");
8         return;
9     }
10
11     int sum = arr[left] + arr[right];
12
13     if (sum == x) {
14         printf("%d\n", arr[left]);
15         printf("%d\n", arr[right]);
16         return;
17     } else if (sum < x) {
18         findPair(left + 1, right, x);
19     } else {
20         findPair(left, right - 1, x);
21     }
22 }
23
24 int main() {
25     int n, x;
26     scanf("%d", &n);
27
28     for (int i = 0; i < n; i++)
29         scanf("%d", &arr[i]);
30
31     scanf("%d", &x);
32
33     findPair(0, n - 1, x);
34
35     return 0;
36 }
```

	Input	Expected	Got	
✓	4	4	4	✓
	2	10	10	
	4			
	8			
	10			
	14			

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 24 September 2025, 8:51 PM**State** Finished**Completed on** Wednesday, 24 September 2025, 8:55 PM**Time taken** 3 mins 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

Write a Program to Implement the Quick Sort Algorithm

**Input Format:**

The first line contains the no of elements in the list-n

The next n lines contain the elements.

**Output:**

Sorted list of elements

**For example:**

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

**Answer:**

```

1 #include <stdio.h>
2
3 void swap(int* a, int* b) {
4     int t = *a;
5     *a = *b;
6     *b = t;
7 }
8
9 int partition(int arr[], int low, int high) {
10    int pivot = arr[high];
11    int i = low - 1;
12
13    for (int j = low; j < high; j++) {
14        if (arr[j] < pivot) {
15            i++;
16            swap(&arr[i], &arr[j]);
17        }
18    }
19
20    swap(&arr[i + 1], &arr[high]);
21    return i + 1;
22 }
23
24 void quickSort(int arr[], int low, int high) {
25    if (low < high) {
26        int pi = partition(arr, low, high);
27        quickSort(arr, low, pi - 1);
28        quickSort(arr, pi + 1, high);
29    }
30 }
31
32 int main() {
33    int n;
34    scanf("%d", &n);
35
36    int arr[n];
37    for (int i = 0; i < n; i++)
38        scanf("%d", &arr[i]);
39
40    quickSort(arr, 0, n - 1);
41
42    for (int i = 0; i < n; i++)
43        printf("%d ", arr[i]);
44
45    return 0;
46 }
```

	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 20 August 2025, 9:15 AM**State** Finished**Completed on** Wednesday, 20 August 2025, 9:16 AM**Time taken** 1 min 40 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

**Question 1** | Correct Mark 1.00 out of 1.00

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

**Output:**

4

**Explanation:**

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

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

```

1 #include <stdio.h>
2
3 int min_coins(int V) {
4     int denominations[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
5     int count = 0;
6     int i = 0;
7     while (V > 0) {
8         count += V / denominations[i];
9         V = V % denominations[i];
10        i++;
11    }
12    return count;
13 }
14
15 int main() {
16     int V;
17     scanf("%d", &V);
18     printf("%d\n", min_coins(V));
19     return 0;
20 }
21

```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 20 August 2025, 9:17 AM**State** Finished**Completed on** Wednesday, 20 August 2025, 9:27 AM**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

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.

Each child  $i$  has a greed factor  $g[i]$ , which is the minimum size of a cookie that the child will be content with; and each cookie  $j$  has a size  $s[j]$ . If  $s[j] \geq g[i]$ , we can assign the cookie  $j$  to the child  $i$ , and the child  $i$  will be content. Your goal is to maximize the number of your content children and output the maximum number.

**Example 1:****Input:**

```
3
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:**

```
1 <= g.length <= 3 * 10^4
0 <= s.length <= 3 * 10^4
1 <= g[i], s[j] <= 2^31 - 1
```

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

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

```
39     int result = find_content_children(g, n, s, m);
40     printf("%d\n", result);
41     free(g);
42     free(s);
43     return 0;
44 }
45 }
```

	Input	Expected	Got	
✓	2	2	2	✓
	1 2			
	3			
	1 2 3			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 2:02 PM**State** Finished**Completed on** Saturday, 1 November 2025, 7:30 PM**Time taken** 19 days 5 hours**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

**Question 1** | Correct Mark 1.00 out of 1.00

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

If he has eaten  $i$  burgers with  $c$  calories each, then he has to run at least  $3^i * c$  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-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 %)

```

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

	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 20 August 2025, 9:23 AM**State** Finished**Completed on** Wednesday, 20 August 2025, 9:37 AM**Time taken** 13 mins 47 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

**Question 1** | Correct Mark 1.00 out of 1.00

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

**Input Format:**

First line specifies the number of elements-n

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>
3
4 int compare_asc(const void *a, const void *b) {
5     int x = *(int*)a;
6     int y = *(int*)b;
7     return x - y;
8 }
9
10 int main() {
11     int n;
12     scanf("%d", &n);
13
14     int *arr = (int*)malloc(n * sizeof(int));
15     for (int i = 0; i < n; i++) {
16         scanf("%d", &arr[i]);
17     }
18
19     // Sort array ascending
20     qsort(arr, n, sizeof(int), compare_asc);
21
22     long long max_sum = 0;
23     for (int i = 0; i < n; i++) {
24         max_sum += (long long)arr[i] * i;
25     }
26
27     printf("%lld\n", max_sum);
28
29     free(arr);
30     return 0;
31 }
32

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	5 2 5 3 4 0	40	40	✓

	Input	Expected	Got	
✓	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Wednesday, 20 August 2025, 9:25 AM**State** Finished**Completed on** Wednesday, 20 August 2025, 9:37 AM**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

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	28
1	
2	
3	
4	
5	
6	

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

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int compare_asc(const void *a, const void *b) {
4     int x = *(int*)a;
5     int y = *(int*)b;
6     return x - y;
7 }
8 int compare_desc(const void *a, const void *b) {
9     int x = *(int*)a;
10    int y = *(int*)b;
11    return y - x;
12 }
13 int main() {
14     int n;
15     scanf("%d", &n);
16     int *A = (int*)malloc(n * sizeof(int));
17     int *B = (int*)malloc(n * sizeof(int));
18     for (int i = 0; i < n; i++) {
19         scanf("%d", &A[i]);
20     }
21     for (int i = 0; i < n; i++) {
22         scanf("%d", &B[i]);
23     }
24     qsort(A, n, sizeof(int), compare_asc);
25     qsort(B, n, sizeof(int), compare_desc);
26     long long sum = 0;
27     for (int i = 0; i < n; i++) {
28         sum += (long long)A[i] * B[i];
29     }
30     printf("%lld\n", sum);
31     free(A);
32     free(B);
33     return 0;
34 }
```

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:39 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:55 PM**Time taken** 15 mins 17 secs**Grade** 10.00 out of 10.00 (100%)

**Question 1** | Correct Mark 10.00 out of 10.00

### 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 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 %)

```

1 #include <stdio.h>
2
3 unsigned long long countWays(int n) {
4     unsigned long long ways[n + 1];
5
6     for(int i = 0; i <= n; i++) {
7         ways[i] = 0;
8     }
9
10    ways[0] = 1;
11
12    for(int i = 1; i <= n; i++) {
13        if(i - 1 >= 0)
14            ways[i] += ways[i - 1];
15        if(i - 3 >= 0)
16            ways[i] += ways[i - 3];
17    }
18
19    return ways[n];
20}
21
22 int main() {
23     int n;
24     scanf("%d", &n);
25
26     printf("%llu\n", countWays(n));
27
28     return 0;
29 }
30
31

```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:42 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:55 PM**Time taken** 12 mins 47 secs**Grade** 10.00 out of 10.00 (100%)

**Question 1** | Correct Mark 10.00 out of 10.00

### Playing with Chessboard:

Ram is given with an  $n \times n$  chessboard with each cell with a monetary value. Ram stands at the  $(0,0)$ , that 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  $n \times n$  chessboard values

#### Output Format

Print Maximum monetary value of the path

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

```
1 #include <stdio.h>
2
3 int max(int a, int b) {
4     return (a > b) ? a : b;
5 }
6
7 int main() {
8     int n;
9     scanf("%d", &n);
10
11    int board[n][n];
12    int dp[n][n];
13
14    for(int i = 0; i < n; i++) {
15        for(int j = 0; j < n; j++) {
16            scanf("%d", &board[i][j]);
17        }
18    }
19
20    dp[0][0] = board[0][0];
21    for(int j = 1; j < n; j++) {
22        dp[0][j] = dp[0][j-1] + board[0][j];
23    }
24    for(int i = 1; i < n; i++) {
25        dp[i][0] = dp[i-1][0] + board[i][0];
26    }
27    for(int i = 1; i < n; i++) {
28        for(int j = 1; j < n; j++) {
29            dp[i][j] = board[i][j] + max(dp[i-1][j], dp[i][j-1]);
30        }
31    }
32
33    printf("%d\n", dp[n-1][n-1]);
34
35    return 0;
36 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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.

[Back to Course](#)

//

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:42 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:56 PM**Time taken** 13 mins 42 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

Given two strings find the length of the common longest subsequence(need not be contiguous) between the two.

Example:

s1: ggtabe

s2: tgatasb

s1	a	g	g	t	a	b	
s2	<b>g</b>	x	<b>t</b>	x	a	y	<b>b</b>

**The length is 4**

Solveing it using Dynamic Programming

For example:

Input	Result
aab	2
azb	

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

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int max(int a, int b) {
5     return (a > b) ? a : b;
6 }
7
8 int longestCommonSubsequence(char* s1, char* s2) {
9     int len1 = strlen(s1);
10    int len2 = strlen(s2);
11
12    int dp[len1 + 1][len2 + 1];
13
14    for (int i = 0; i <= len1; i++) {
15        for (int j = 0; j <= len2; j++) {
16            if (i == 0 || j == 0)
17                dp[i][j] = 0;
18            else if (s1[i - 1] == s2[j - 1])
19                dp[i][j] = 1 + dp[i - 1][j - 1];
20            else
21                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
22        }
23    }
24
25    return dp[len1][len2];
26 }
27
28 int main() {
29     char s1[1001], s2[1001];
30     scanf("%s %s", s1, s2);
31
32     int result = longestCommonSubsequence(s1, s2);
33     printf("%d\n", result);
34
35     return 0;
36 }
37

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:43 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:56 PM**Time taken** 13 mins 41 secs**Marks** 1.00/1.00**Grade** **10.00** out of 10.00 (**100%**)

**Question 1** | Correct Mark 1.00 out of 1.00

**Problem statement:**

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 longestNonDecreasingSubsequence(int arr[], int n) {
8     int dp[n];
9
10    for(int i = 0; i < n; i++)
11        dp[i] = 1;
12
13    for(int i = 1; i < n; i++) {
14        for(int j = 0; j < i; j++) {
15            if(arr[j] <= arr[i])
16                dp[i] = max(dp[i], dp[j] + 1);
17        }
18    }
19
20    int ans = dp[0];
21    for(int i = 1; i < n; i++) {
22        if(dp[i] > ans)
23            ans = dp[i];
24    }
25
26    return ans;
27
28
29 }
30
31 int main() {
32     int n;
33     scanf("%d", &n);
34     int arr[n];
35     for(int i = 0; i < n; i++) {
36         scanf("%d", &arr[i]);
37     }
38
39     printf("%d\n", longestNonDecreasingSubsequence(arr, n));
40     return 0;
41 }
42

```

	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:43 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:56 PM**Time taken** 13 mins 36 secs**Marks** 1.00/1.00**Grade** **4.00** out of 4.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

Find Duplicate in Array.

Given 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 1 2 3 4	

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

```

1 #include <stdio.h>
2 int findDuplicate(int arr[], int n) {
3     int slow = arr[0];
4     int fast = arr[0];
5     do {
6         slow = arr[slow];
7         fast = arr[arr[fast]];
8     } while (slow != fast);
9     slow = arr[0];
10    while (slow != fast) {
11        slow = arr[slow];
12        fast = arr[fast];
13    }
14
15    return slow;
16 }
17
18 int main() {
19     int n;
20     scanf("%d", &n);
21     int arr[n];
22
23     for(int i = 0; i < n; i++) {
24         scanf("%d", &arr[i]);
25     }
26
27     int duplicate = findDuplicate(arr, n);
28     printf("%d\n", duplicate);
29
30     return 0;
31 }
32

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

 **BINESH KARTHIK S 2024-CSE** ▾**B2****Started on** Monday, 13 October 2025, 1:47 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:57 PM**Time taken** 9 mins 20 secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

Find Duplicate in Array.

Given 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 1 2 3 4	

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

```

1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &arr[i]);
10    }
11
12    int duplicate = -1;
13    for (int i = 0; i < n; i++) {
14        for (int j = i + 1; j < n; j++) {
15            if (arr[i] == arr[j]) {
16                duplicate = arr[i];
17                break;
18            }
19        }
20        if (duplicate != -1)
21            break;
22    }
23
24    if (duplicate != -1)
25        printf("%d\n", duplicate);
26    else
27        printf("No duplicate found\n");
28
29    return 0;
30}
31

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)

 **BINESH KARTHIK S 2024-CSE** ▾**B2****Started on** Monday, 13 October 2025, 1:47 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:57 PM**Time taken** 9 mins 19 secs**Marks** 1.00/1.00**Grade** **30.00** out of 30.00 (**100%**)

**Question 1** | Correct Mark 1.00 out of 1.00

Find the intersection of two sorted arrays.

OR in other words,

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

**Input Format**

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

- Line 1 contains N1, followed by N1 integers of the first array
- 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	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

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

```

1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T);
6
7     while (T--) {
8         int n1, n2;
9         scanf("%d", &n1);
10        int arr1[n1];
11        for (int i = 0; i < n1; i++) {
12            scanf("%d", &arr1[i]);
13        }
14
15        scanf("%d", &n2);
16        int arr2[n2];
17        for (int i = 0; i < n2; i++) {
18            scanf("%d", &arr2[i]);
19        }
20
21        int i = 0, j = 0;
22        int found = 0.

```

```

23 ↓
24 ↓
25
26 ↓
27
28 ↓
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
    ... ,  

    while (i < n1 && j < n2) {  

        if (arr1[i] < arr2[j]) {  

            i++;  

        } else if (arr1[i] > arr2[j]) {  

            j++;  

        } else {  

            printf("%d ", arr1[i]);  

            found = 1;  

            i++;  

            j++;  

        }  

    }  

    if (found)  

        printf("\n");  

    else  

        printf("No common elements\n");  

}  

return 0;
}

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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.

[Back to Course](#)

 **BINESH KARTHIK S 2024-CSE** ▾**B2****Started on** Monday, 13 October 2025, 1:48 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:57 PM**Time taken** 9 mins 33 secs**Marks** 1.00/1.00**Grade** **30.00** out of 30.00 (**100%**)

**Question 1** | Correct Mark 1.00 out of 1.00

Find the intersection of two sorted arrays.

OR in other words,

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

**Input Format**

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

- Line 1 contains N1, followed by N1 integers of the first array
- 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	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

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

```

1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T);
6
7     while (T--) {
8         int n1, n2;
9         scanf("%d", &n1);
10        int arr1[n1];
11        for (int i = 0; i < n1; i++) {
12            scanf("%d", &arr1[i]);
13        }
14
15        scanf("%d", &n2);
16        int arr2[n2];
17        for (int i = 0; i < n2; i++) {
18            scanf("%d", &arr2[i]);
19        }
20
21        int i = 0, j = 0;
22        int found = 0.

```

```

23 ↓
24 ↓
25
26 ↓
27
28 ↓
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
    ... ,  

    while (i < n1 && j < n2) {  

        if (arr1[i] < arr2[j]) {  

            i++;  

        } else if (arr1[i] > arr2[j]) {  

            j++;  

        } else {  

            printf("%d ", arr1[i]);  

            found = 1;  

            i++;  

            j++;  

        }  

    }  

    if (found)  

        printf("\n");  

    else  

        printf("No common elements\n");  

}  

return 0;
}

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:48 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:57 PM**Time taken** 9 mins 33 secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

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

Input Format:

First Line 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
1 3 5	
4	

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

```

1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &arr[i]);
10    }
11
12    int k;
13    scanf("%d", &k);
14
15    int i = 0, j = 1;
16    int found = 0;
17    while (i < n && j < n) {
18        int diff = arr[j] - arr[i];
19
20        if (diff == k && i != j) {
21            found = 1;
22            break;
23        }
24        else if (diff < k) {
25            j++;
26        }
27        else {
28            i++;
29        }
30    }
31
32    printf("%d\n", found);
33
34    return 0;
35}
36

```

	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.

[Back to Course](#)

BINESH KARTHIK S 2024-CSE ▾**B2****Started on** Monday, 13 October 2025, 1:48 PM**State** Finished**Completed on** Monday, 13 October 2025, 1:58 PM**Time taken** 9 mins 8 secs**Marks** 1.00/1.00**Grade** 4.00 out of 4.00 (100%)

**Question 1** | Correct Mark 1.00 out of 1.00

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

Input Format:

First Line 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
1 3 5	
4	

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

```

1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &arr[i]);
10    }
11
12    int k;
13    scanf("%d", &k);
14
15    int i = 0, j = 1;
16    int found = 0;
17
18    while (i < n && j < n) {
19        int diff = arr[j] - arr[i];
20
21        if (diff == k && i != j) {
22            found = 1;
23            break;
24        }
25        else if (diff < k) {
26            j++;
27        }
28        else {
29            i++;
30        }
31    }
32
33    printf("%d\n", 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.

[Back to Course](#)