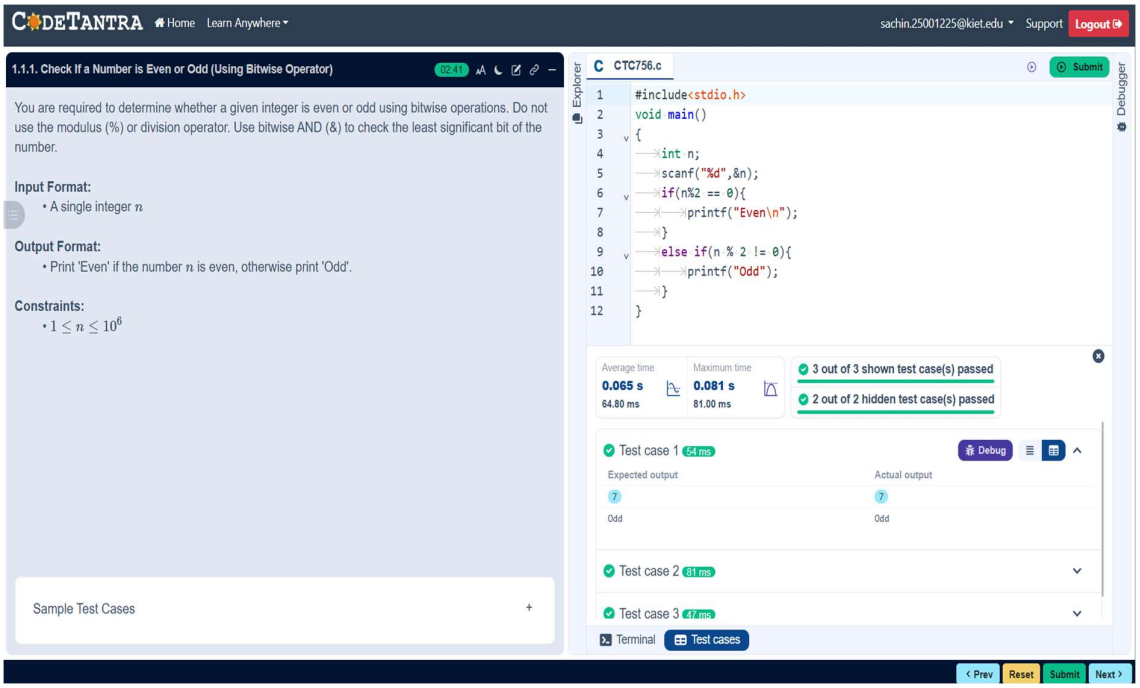
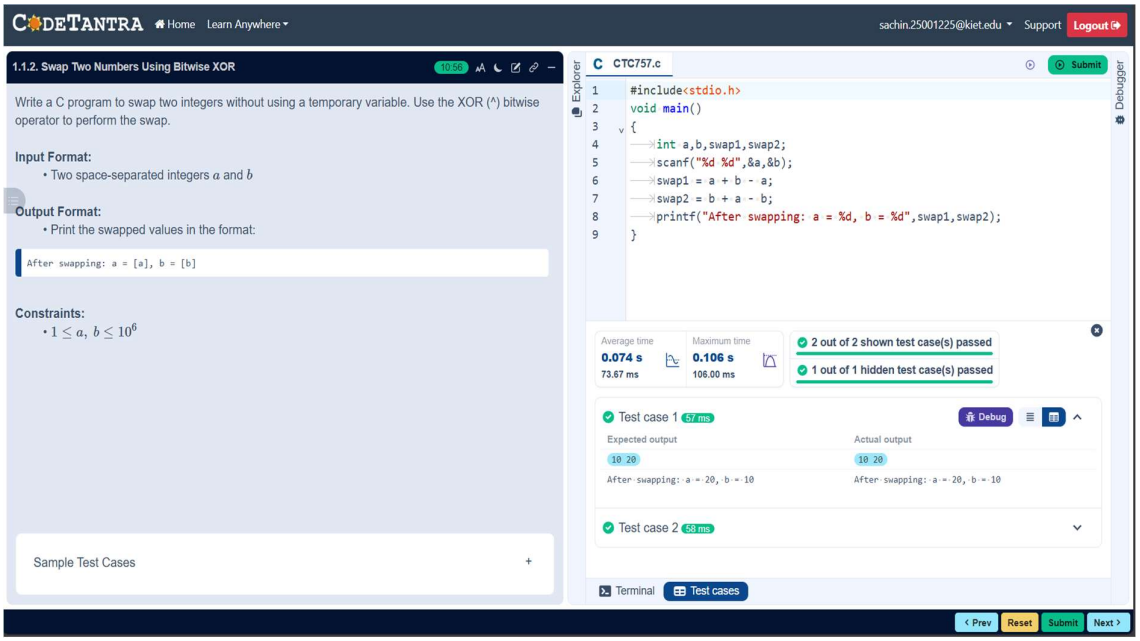

CODETANTRA
Home
Learn Anywhere
sachin.25001225@kiet.edu
Support
Logout
1.1.1. Check If a Number is Even or Odd (Using Bitwise Operator)
02:41
You are required to determine whether a given integer is even or odd using bitwise operations. Do not use the modulus (%) or division operator. Use bitwise AND (&) to check the least significant bit of the number.
Input Format:
• A single integer n
Output Format:
• Print 'Even' if the number n is even, otherwise print 'Odd'.
Constraints:
• $1 \leq n \leq 10^6$
Sample Test Cases
Explorer
CTC756.c
Submit
Debugger
#include<stdio.h>
void main()
{
int n;
scanf("%d",&n);
if(n%2 == 0){
printf("Even\n");
}
else if(n % 2 != 0){
printf("Odd");
}
}
Average time
0.065 s
64.80 ms
Maximum time
0.081 s
81.00 ms
3 out of 3 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1 64 ms
Debug
Expected output
7
Odd
Actual output
7
Odd
Test case 2 81 ms
Test case 3 47 ms
Terminal
Test cases
Prev
Reset
Submit
Next


CODETANTRA
Home
Learn Anywhere
sachin.25001225@kiet.edu
Support
Logout
1.1.2. Swap Two Numbers Using Bitwise XOR
10:56
Write a C program to swap two integers without using a temporary variable. Use the XOR (^) bitwise operator to perform the swap.
Input Format:
• Two space-separated integers a and b
Output Format:
• Print the swapped values in the format:
After swapping: a = [a], b = [b]
Constraints:
• $1 \leq a, b \leq 10^6$
Sample Test Cases
Explorer
CTC757.c
Submit
Debugger
#include<stdio.h>
void main()
{
int a,b,swap1,swap2;
scanf("%d %d",&a,&b);
swap1 = a + b - a;
swap2 = b + a - b;
printf("After swapping: a = %d, b = %d",swap1,swap2);
}
Average time
0.074 s
73.67 ms
Maximum time
0.106 s
106.00 ms
2 out of 2 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1 57 ms
Debug
Expected output
10 20
After swapping: a = 20, b = 10
Actual output
10 20
After swapping: a = 20, b = 10
Test case 2 58 ms
Terminal
Test cases
Prev
Reset
Submit
Next