

How To Complete Online Quiz Tests

1 Navigate to <https://campus.binarykeeda.com>

- The total duration of the test is
- The test consists of
- Sections may be of type **Quiz** or **Coding**.
- Each **Quiz** section includes MCQs, MSQs, and short answer questions.
- Each **Coding** section presents a problem statement — you must solve it using the provided code editor.
- **Tab change monitoring** is enabled. Switching tabs may negatively impact your result.
- **Do not close or refresh the browser tab once the test starts.** Doing so may lead to data loss or disqualification.
- **Plagiarism is strictly prohibited.** All code submissions must be your own.

➤ In **Quiz sections**, question status is color-coded:

Attempted (Done) Seen but not attempted Unseen / Not visited Current Question

- The test will be **auto-submitted** once the time runs out. Please track the on-screen timer.
- Once a section is submitted, you may **not be able to return** to it. Double-check before submitting.
- **Do not use any external help** like ChatGPT, online IDEs, or discuss answers with anyone.
- **Use a laptop or desktop.** Mobile devices are not recommended.
- **Ensure a stable internet connection.** Disruptions may affect your test progress.

By clicking "Start Test", you agree to follow the rules above. Any violation may lead to test cancellation.

2 Click "Start Test"

- Each **Quiz** section includes MCQs, MSQs, and short answer questions.
- Each **Coding** section presents a problem statement — you must solve it using the provided code editor.
- **Tab change monitoring** is enabled. Switching tabs may negatively impact your result.
- **Do not close or refresh the browser tab once the test starts.** Doing so may lead to data loss or disqualification.
- **Plagiarism is strictly prohibited.** All code submissions must be your own.
- In **Quiz sections**, question status is color-coded:
 - Attempted (Done)
 - Seen but not attempted
 - Unseen / Not visited
 - Current Question
- The test will be **auto-submitted** once the time runs out. Please track the on-screen timer.
- Once a section is submitted, you may **not be able to return** to it. Double-check before submitting.
- **Do not use any external help** like ChatGPT, online IDEs, or discuss answers with anyone.
- **Use a laptop or desktop.** Mobile devices are not recommended.
- **Ensure a stable internet connection.** Disruptions may affect your test progress.

By clicking "Start Test", you agree to follow the rules above. Any violation may lead to test cancellation.

Start Test

3 Click on Section with Blue background

BinaryKeeda

?

Aptitude

Quiz

20 Questions for 30 minutes

↗

Computer Science Basics

Quiz

10 Questions for 20 minutes

🔒

Autmata

Coding

2 Questions for 45 minutes

🔒

Automata Fix

Coding

3 Questions for 45 minutes

🔒

4 Attempt Question



29 min | 56 sec

Q1. The average of 20 numbers is 35. Later it is found that a number 65 was misread as 56. What is the correct average?

- 35.45
- 35.25
- 34.75
- 35.50



Clear choice

Question Navigator

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Previous Next

5 Click "Submit" to submit section

- 35.25
- 34.75
- 35.50

Clear choice

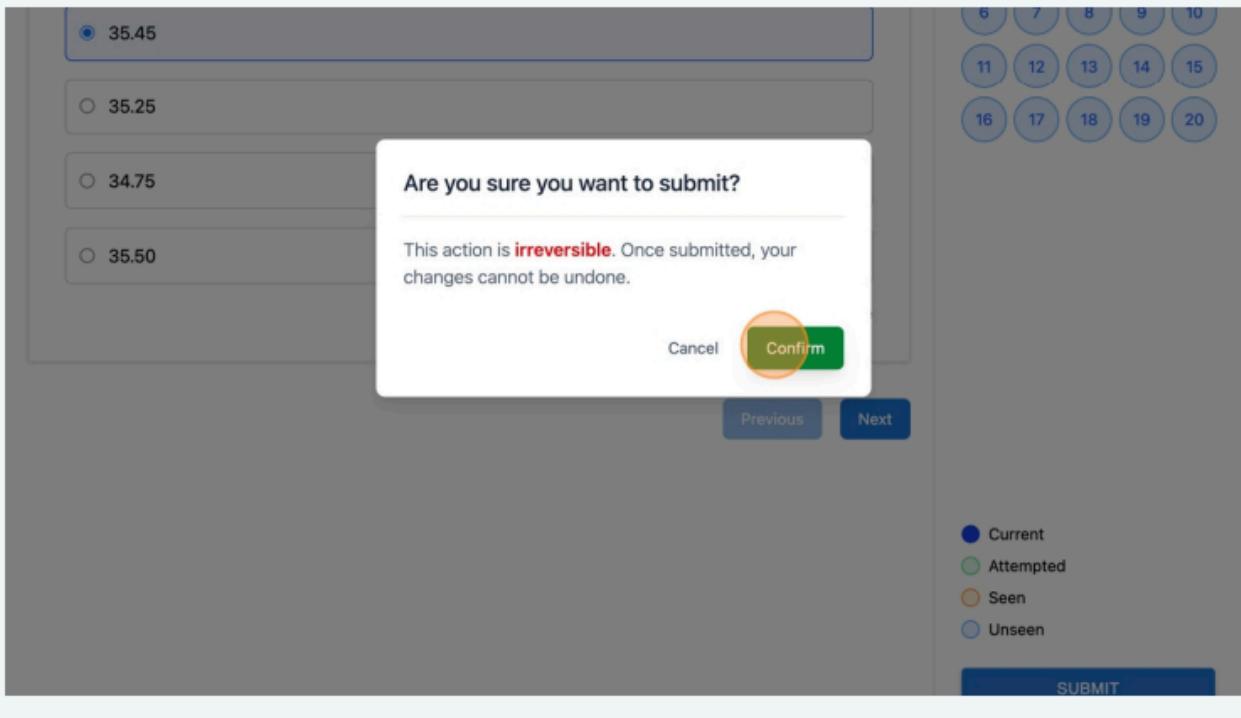
Previous Next

11	12	13	14	15
16	17	18	19	20

- Current
- Attempted
- Seen
- Unseen

SUBMIT

6 Click "Confirm"



7 Click on another section now

Aptitude

Quiz

20 Questions for 30 minutes



Computer Science Basics

Quiz

10 Questions for 20 minutes



Autmata

Coding

2 Questions for 45 minutes



Automata Fix

Coding

3 Questions for 45 minutes



Note : All the sections are mandatory to attempt then only the submmision would be counted

8 This is Coding interface , Click on arrow button to change section

Explanation

To solve this problem, we start by converting the number 45 into its binary form.

The binary representation of 45 is 101101, which uses 6 bits — the minimum required to represent it without padding.

Next, we flip each bit:

1 becomes 0

0 becomes 1

So, flipping 101101 gives us 010010.

Now, we convert the flipped binary 010010 back to its decimal form, which is 18.

...

Testcases >

Input:

1

Expected Output:

0

Input:

```
8  public class Main {  
9      public static void main(String[] args) {  
10         Scanner sc = new Scanner(System.in);  
11         int N = sc.nextInt();  
12  
13         Solution sol = new Solution();  
14         int result = sol.flipBits(N);  
15  
16         System.out.println(result);  
17     }  
18 }  
19  
20
```

< 1/2 >

9 Click "Submit Section" to submit the complete section (note: individual code submission not accepted , complete writing all codes and than submit)



Sid and Rid's Subarray Game

You are given an array A of N integers which may include both positive and negative numbers. Two friends, Sid and Rid, decide to play a game: Sid loves positive numbers. He creates a new array S consisting only of the positive numbers from A, preserving their original order. Rid works on the original array A. Each of them does the following: Finds the maximum sum subarray from their respective array. Checks whether their resulting maximum sum subarray contains the majority element (an element that occurs more than $\text{floor}(N/2)$ times in the original array A). You are to write a program that: Computes the maximum subarray sum for both Sid and Rid. Returns whether the majority element (if it exists) is present in each of their maximum subarrays.

Constraints

- $1 \leq N \leq 10^5$
- Array A may contain both positive and negative integers

Sample testcases

Input:

9 1 -2 3 4 -1 2 1 -5 4

...

Testcases >

```
00:44:55 JAVA Run Submit Section  
1  import java.util.*;  
2  
3  class Solution {  
4      public String evaluateGame(int[] A) {  
5          int n = A.length;  
6          Integer majority = getMajority(A);  
7  
8          // Rid: max subarray on A  
9          Result ridRes = kadane(A);  
10         boolean ridHas = majority != null && contains(ridRes.subarray, majority);  
11  
12         // Sid: max subarray on positives  
13         List<Integer> posList = new ArrayList<>();  
14         for (int x : A) if (x > 0) posList.add(x);  
15         int[] S = posList.stream().mapToInt(i -> i).toArray();  
16  
17         Result sidRes = kadane(S);  
18         boolean sidHas = majority != null && contains(sidRes.subarray, majority);  
19  
20         return "Rid: " + ridRes.maxSum + " " + (ridHas ? "Yes" : "No")  
21         | "Sid: " + sidRes.maxSum + " " + (sidHas ? "Yes" : "No");  
22     }  
23  
24     private boolean contains(int[] arr, int val) {  
25         for (int x : arr) if (x == val) return true;  
26         return false;  
27     }  
28 }
```


12 After final submission your screen would look like this

