

Crack The Code_

"Where logic meets code – let's crack it!!"

Coordinator- Prof. Dr. Soumen Paul
Prof. Manasija Bhattacharya
Prof. Moumita Ghosh

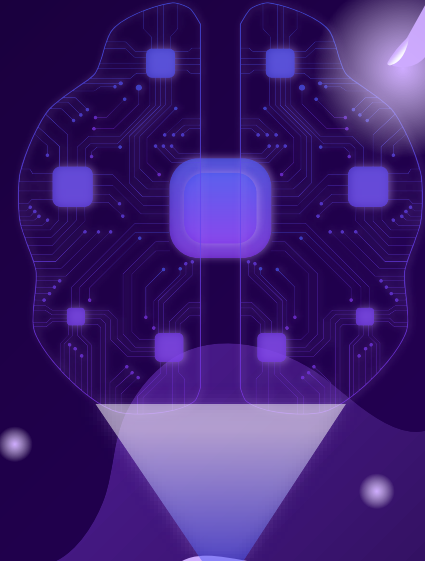


Table of contents_

01

Placement Drive
in HIT

03

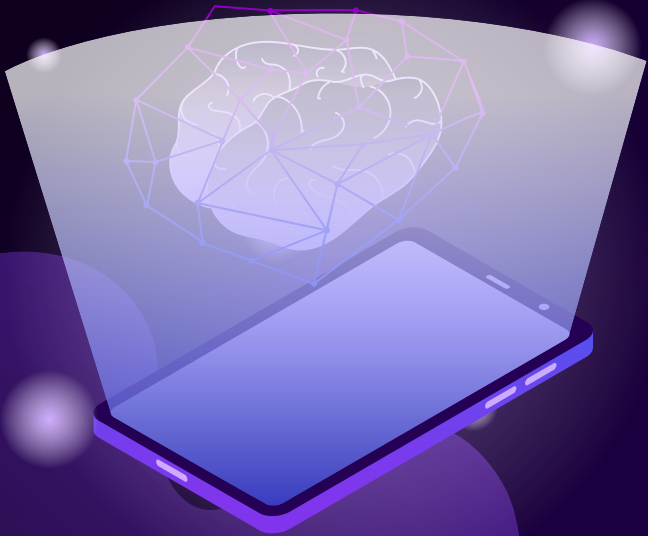
DSA Round

02

Aptitude Round

04

Alumni Session



01

Placement drive
in HIT

Placement Process_

- **Preparation:** Training sessions, resume building, and mock interviews.
- **Company Registration:** T&P cell invites companies; students apply based on eligibility.
- **Aptitude & Coding Test** – Logical, technical MCQs & DSA problems.
- **Group Discussion (if applicable)** – Communication & teamwork assessment.
- **Technical Interview** – DSA, OOPs, DBMS, OS, projects, and core subjects.
- **HR Interview** – Personality, communication, and company fit.
- **Offer & Joining:** Selected students receive offer letters; some start with internships.
- **Backup Plans:** Off-campus applications via LinkedIn, job portals, and referrals.

On campus and Off campus_

◆ Feature



On-Campus Placement



Off-Campus Placement

Where it happens

At your college (or via placement cell)

Outside college (job portals, LinkedIn, etc.)

Who organizes it

College's Training & Placement Cell

You apply directly / through referrals

Eligibility

Only for students of that college

Open to all eligible candidates (fresher/exp)

Competition

Limited (only batchmates)

High (students from all over India)

Company access

Limited to companies visiting the college

Much broader (startups, MNCs, global firms)

PBCs and SBCs

Product-Based Companies (PBCs)

These companies **build and sell their own products** — like apps, platforms, or tools. Their main business = **the product** itself.

- ◆ **Examples:**

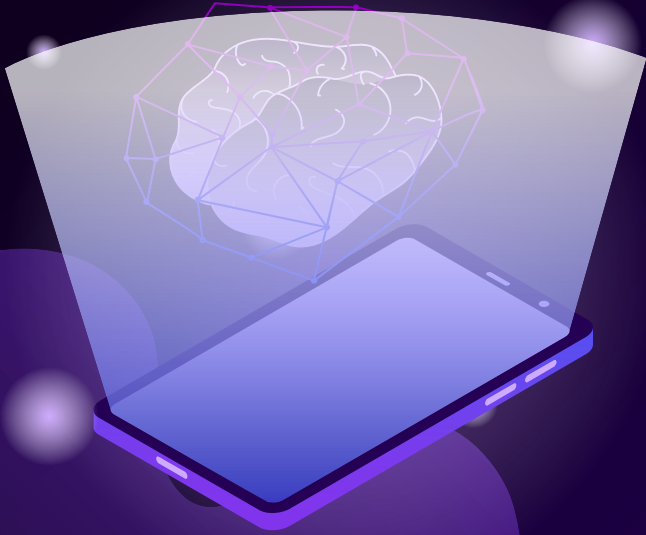
- Google
- Microsoft
- Adobe
- Amazon

Service-Based Companies (SBCs)

These companies provide **IT services to other businesses**. You work on **client projects**, not your own product.

- ◆ **Examples:**

- TCS
- Infosys
- Wipro
- Cognizant



02

Aptitude Round_

Aptitude Round_

- **Quantitative Aptitude** – Questions on profit-loss, percentages, time-speed-distance, permutations, etc.
- **Logical Reasoning** – Puzzles, seating arrangements, number series, syllogisms, etc.
- **Verbal Ability** – Grammar, reading comprehension, sentence correction, etc.
- **Technical MCQs** – Programming, data structures, OS, DBMS, and CS fundamentals.

Quantitative Aptitude_

The price of a laptop is increased by 10% and then by 15%. What is the overall percentage increase in price?

- A) 25%
- B) 26.5%
- C) 27%
- D) 24.5%

Quantitative Aptitude_

If $36 : 81 :: X : 63$, find the value of X.


A) 28

B) 32

C) 42

D) 49

Logical Reasoning_

Three statements are given, followed by three conclusions numbered I, II, and III. Assuming the statements to be true, even if they seem to be at variance with commonly known facts, decide which of the conclusions logically follow(s) from the statements. 

Statements:

- All pages are diaries.
- Some diaries are books.
- All books are notebooks.

Conclusions:

- I. No page is a notebook.
- II. All pages are notebooks.
- III. All diaries being notebooks is a possibility.

Verbal Ability_

Identify the grammatically correct phrase to replace the underlined portion in the following sentence:

"The man to who I sold my house was a cheat."

Options:

- (A) to whom I sell
- (B) to who I sell
- (C) who was sold to
- (D) to whom I sold

Technical MCQs_

What is the output of the following C code snippet?

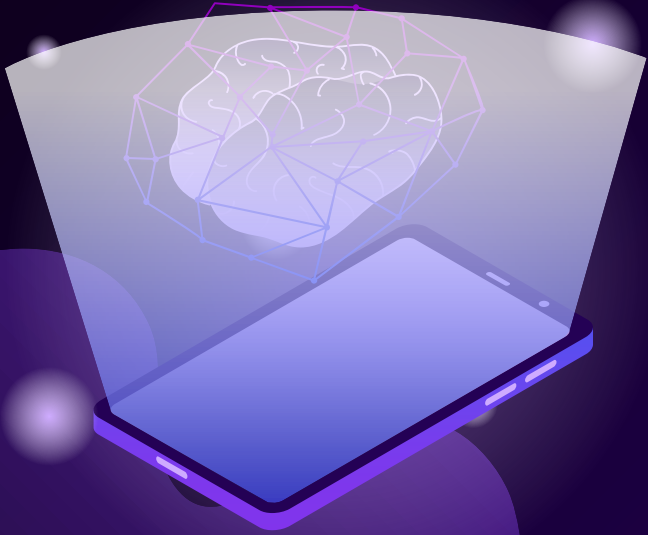
C



```
int main() {  
    int x = 10;  
    printf("%d", x++ + ++x);  
    return 0;  
}
```

Options:


- (A) 20
- (B) 21
- (C) 22
- (D) 23



03

DSA Round_

Sum of Array_



Problem

Editorial

Submissions

Comments

Sum of Array

Difficulty: Basic Accuracy: 74.98% Submissions: 125K+ Points: 1 Average Time: 20m

You are given an integer array `arr[]`. The task is to find the **sum** of it.

Examples:

Input: `arr[] = [1, 2, 3, 4]`
Output: 10
Explanation: $1 + 2 + 3 + 4 = 10$.

Input: `arr[] = [1, 3, 3]`
Output: 7
Explanation: $1 + 3 + 3 = 7$.

Constraints:

$1 \leq \text{arr.size} \leq 10^5$
 $1 \leq \text{arr}[i] \leq 10^4$

[Try more examples](#)

Expected Complexities

Topic Tags

Related Articles

[Report An Issue](#)

C++ (g++ 5.4)

Start Timer


```
1 // } Driver Code Ends
2
3 // User function template for C++
4 class Solution {
5 public:
6     //
7     int arraySum(int arr,int size) {
8         int sum = 0;
9         for (int i = 0; i < size; i++) {
10             sum += arr[i];
11         }
12         return sum;
13     }
14 };
15
16
17
18
19
20
21
22
```

Custom Input


Compile & Run


Submit


Largest Element in Array_



Problem

 Editorial

 Submissions

 Comments

Largest Element in Array

Difficulty: Basic Accuracy: 67.48% Submissions: 456K+ Points: 1 Average Time: 20m

Given an array `arr[]`. The task is to find the **largest** element and return it.

Examples:

Input: `arr[] = [1, 8, 7, 56, 90]`
Output: 90
Explanation: The largest element of the given array is 90.

Input: `arr[] = [5, 5, 5, 5]`
Output: 5
Explanation: The largest element of the given array is 5.

Input: `arr[] = [10]`
Output: 10
Explanation: There is only one element which is the largest.

Constraints:

$1 \leq \text{arr.size()} \leq 10^6$
 $0 \leq \text{arr}[i] \leq 10^6$

Try more examples

C (gcc 5.4) Start Timer

```
1 // } Driver Code Ends
10
11 // User function Template for C
12
13 // Function to check if the array is sorted
14 int largest(int arr[], int n) {
15     int max = arr[0];
16     for (int i = 1; i < n; i++) {
17         if (arr[i] > max) {
18             max = arr[i];
19         }
20     }
21     return max;
22 }
23
24
25 // } Driver Code Ends
```

Custom Input Compile & Run Submit

Reverse String_

Problem List

344. Reverse String

Easy Topics Companies Hint

Write a function that reverses a string. The input string is given as an array of characters `s`.
You must do this by modifying the input array **in-place** with **$O(1)$** extra memory.

Example 1:
Input: `s = ["h","e","l","l","o"]`
Output: `["o","l","l","e","h"]`

Example 2:
Input: `s = ["H","a","n","n","a","h"]`
Output: `["h","a","n","n","a","H"]`

Constraints:

- $1 \leq s.length \leq 10^5$
- `s[i]` is a **printable ascii character**.

8.9K 251 61 Online

Code

```
1 void reverseString(char* s, int sSize) {  
2     for(int i=0;i<(sSize/2);i++){  
3         char temp=s[i];  
4         s[i]=s[(sSize-1)-i];  
5         s[(sSize-1)-i]=temp;  
6     }  
7 }
```

Ln 6, Col 6 | Saved

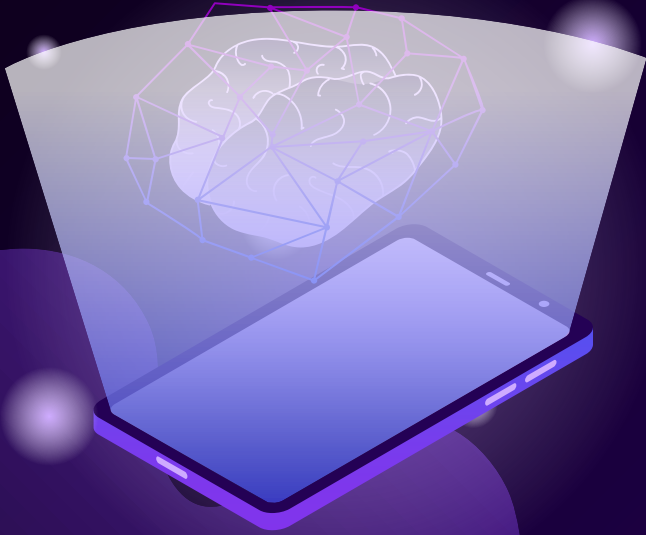
Run Submit

Testcase Test Result

Case 1 Case 2 +

s =
["h","e","l","l","o"]

</> Source



03

Alumni Session_

Our Alumni_



Chandan Singha

**System Engineer @TCS
Batch - 2024**

The background is a solid dark blue. It is decorated with numerous light blue circles of varying sizes, some of which have a soft glow. Two stylized, light blue hands are positioned on the left and right sides, with their index fingers pointing towards the center. The hands are simple in design, with visible fingers and thumbs. In the center of the image, the text "Thank You_" is written in a bold, white, sans-serif font. The text is slightly offset to the right, with the underscore at the end of the word "You".

Thank You_