



WELCOME TO
hackBits

STARTS AT 7:05PM

blue	white	yellow	blue	red
green	yellow	blue	green	black
yellow	red	blue	green	yellow
red	yellow	red	white	green
blue	green	red	yellow	black
blue	black	blue	black	white
black	yellow	white	blue	green
red	black	red	yellow	green
red	yellow	blue	white	blue
red	yellow	blue	yellow	green

Mentimeter uses cookies

We do this to ensure that our website works well and provide you with relevant content and marketing campaigns. We also want to improve our features and research new ones.

Below you can accept all cookies or click 'Cookie Preferences' to choose which ones you want.

If you want to know precisely why we use cookies and other similar technologies or how you can withdraw consent, simply visit our [Cookie Policy](#).

Accept All

Reject All



WHY INTERVIEW?



INTERVIEW PROCESS



RESUME AND APPLICATION REVIEW

The company will review your resume and application materials to assess your fit for the role and to identify potential candidates for the technical interview.



PHONE SCREEN

You may be asked to participate in a phone screen with a recruiter or hiring manager to discuss your background and experience. This is an opportunity for the company to learn more about you and to see if you are a good fit for the role.



TECHNICAL INTERVIEW

If you pass the phone screen, you will be invited to participate in a technical interview. This may be conducted in person or remotely and may involve solving technical problems, writing code, or answering technical questions.



ON-SITE INTERVIEW

If you pass the technical interview, you may be invited to participate in an on-site interview, which typically involves a series of interviews with various team members. These interviews may include technical interviews, as well as behavioral and cultural fit interviews.



OFFER AND ACCEPTANCE

If the company is interested in hiring you, they will extend a job offer. You may be given the opportunity to negotiate the terms of the offer, including salary and benefits.

If you accept the offer, you will be asked to complete any remaining onboarding tasks, such as signing paperwork and completing any required training.



**WELL, WHAT
CAN I BE
ASKED?**



CODING CHALLENGES

1. You may be asked to write code to solve a specific problem or implement a certain feature. This can be done on a whiteboard, on paper, or using a collaborative online coding platform.
2. Example questions may include:
 - Write a function to reverse a linked list
 - Implement a stack using an array
 - Write a SQL query to find the top 5 most popular products in a store



SYSTEM DESIGN QUESTIONS

1. You may be asked to design a system or component of a system, such as a database or network. These questions may involve discussing trade-offs and limitations of different design choices.
2. Example questions may include:
 - Design a distributed system for a social networking platform
 - Design a database schema for a messaging application
 - Design an API for a weather forecasting service



TECHNICAL QUESTIONS

1. You may be asked questions about specific technologies, tools, or concepts that are relevant to the role you are applying for. This may include questions about data structures, algorithms, networking, or software design principles.
2. Example questions may include:
 - Explain the difference between a hash table and a linked list
 - Describe how TCP/IP works
 - Explain the difference between a class and an object in object-oriented programming



PROBLEM-SOLVING EXERCISES

1. You may be asked to solve a technical problem or debug code. This can involve working through the problem step by step and explaining your thought process.
2. Example questions may include:
3. Find and fix the bug in this piece of code
4. Optimize this algorithm for better performance
5. Design a system to handle high volume traffic on a website



CODING INTERVIEW PREPARATION



RESEARCH

Familiarize yourself with the technologies and tools used at the company. Research the company's technology stack and the specific technologies and tools that may be used in the role you are applying for.

Research the company and the role you are applying for. Familiarize yourself with the company's mission, products, and culture, as well as the specific responsibilities and requirements of the role you are applying for.



PRACTICE

Practice coding challenges and problem-solving exercises. There are many websites and resources that offer practice coding challenges, such as HackerRank, LeetCode, and CodeForces.

Review key concepts and best practices for coding. Books such as "Cracking the Coding Interview" and "Elements of Programming Interviews" can be helpful for reviewing these concepts.



REVIEW

Review your resume and projects. Make sure your resume accurately reflects your skills and experience, and be prepared to discuss your past projects in detail. Practice explaining your responsibilities and the technologies you used on each project

Practice communicating your thought process and explaining your solutions to others. This can involve explaining your solutions to a friend or mentor, or working on problems with a study group.



DS&A TOOLKIT



PLATFORMS

- Leetcode
- CodeSignal
- HackerRank
- Codeforces
- Codechef



BOOKS

- Elements of Programming Interviews
- Cracking the coding interview
- System Design Interview : An Insider's Guide



TOPICS

- Strings / Array
- Pointers
- Dynamic Programming
- Graph Search
- Binary Trees / Linked Lists
- and more ...

CONDUCT AN INTERVIEW



TIPS

- Ask clarification questions
- Explain your thought process
- Write pseudocode
- Go for the brute-force solutions first!
- Keep calm!



**ENOUGH
LISTENING.**

TIME TO CODE.



GROUPS

- Make groups
- Introduce yourselves, talk about your favorite artist and then tell us so we play some music while you work



Q1

Given an integer array `nums`, return `true` if any value appears at least twice in the array, and return `false` if every element is distinct.

Example 1:

Input: `nums = [1,2,3,1]`

Output: `true`

Example 2:

Input: `nums = [1,2,3,4]`

Output: `false`

Example 3:

Input: `nums = [1,1,1,3,3,4,3,2,4,2]`

Output: `true`



SOLUTION

Input: nums = [1,2,3,1]

Output: true

```
def containsDuplicate(self, nums):  
    array = []  
    for num in nums:  
        if num in array:  
            return True  
        array.append(num)  
  
    return False
```



Q2 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



SOLUTION

```
def majorityElement(self, nums):  
    hashmap = {}  
    for num in nums:  
        if num in hashmap:  
            hashmap[num] += 1  
        else:  
            hashmap[num] = 1  
  
    for key in hashmap:  
        if hashmap[key] > (len(nums) / 2):  
            return key
```



**SHOW US HOW
IT'S DONE.**



ACTIVITY

- 3 Questions that get progressively harder.
- 30 minutes.
- Communicate between your teammates, and find the solution the fastest.

GOOD LUCK!



1 TWO SUM

9. PALINDROME NUMBER

242. VALID ANAGRAM



SOLUTION

```
def twoSumBrute(nums, target):  
    for i in range(len(nums)):  
        for j in range(i+1, len(nums)):  
            if nums[i] + nums[j] == target:  
                return [i, j]
```



SOLUTION

```
function isPalindrome(x) {  
  // Convert the integer to a string  
  const str = String(x);  
  
  // Reverse the string  
  const reverseStr = str.split("").reverse().join("");  
  // Check if the original and reversed strings are equal  
  if (str === reverseStr) {  
    return true; // It's a palindrome  
  } else {  
    return false; // It's not a palindrome  
  }  
}
```



SOLUTION

```
def isAnagram(self, s, t):  
    hashmap1 = {}  
    hashmap2 = {}  
    for c in s:  
        if c not in hashmap1:  
            hashmap1[c] = 1  
        else:  
            hashmap1[c] += 1  
  
    for c in t:  
        if c not in hashmap2:  
            hashmap2[c] = 1  
        else:  
            hashmap2[c] += 1  
  
    return hashmap1 == hashmap2
```

CIS⁵UFT

1V1



ONE LAST QUESTION.

- SELECT ONE PERSON TO CODE.
- THE REST OF YOU ARE GOING TO BE HELPING HIM FORMULATE THE SOLUTION IN ANY PROGRAMMING LANGUAGE.
- FIRST ONE THAT SOLVES IT CORRECTLY WINS.

ARE YOU READY?



THANK YOU

JOIN US *and bring your friends :)*

EVERY **MONDAY 7:00 PM** AT **ILC S231**



Feedback



Resource GitHub

TIPS

**CHANGE BACKGROUND COLOR TO
THE SESSIONS THEME
MAKE SURE EVERYTHING IS UNIFORM
KEEP ALIGNMENT CENTERED
DONT CHANGE FONT SIZES
DONT CHANGE TOO MANY COLORS
DONT ADD EVERYTHING ON ONE
SLIDE, MAKE MULTIPLE**



TITLE SLIDE

CSOFT

SUBTITLE

PARAGRAPH



TITLE

CSOFT

stuff

stuff

TITLE

stuff

stuff

stuff



TITLE

Subtitle



RESOURCE HUB





Write us Feedback through our website

