Task: Career Guidance 2

Technical Interview Assessments

www.hyperiondev.com

Introduction

Welcome to the Career Guidance & Support module!

Congratulations on making it this far! You have reached the end of your bootcamp and have acquired truly valuable skills which will help you excel in your career as a coder. Now that you have built your CV and portfolio, in this task we will be looking at some sample technical interview assessments and take-home tests in preparation for your next interview!

CONNECT WITH YOUR MENTOR

You don't have to take our courses alone! This course has been designed to be taken with an online mentor that marks your submitted code by hand and supports you in achieving your career goals on a daily basis.

To access this mentor support, simply navigate to www.hyperiondev.com/support.

While you may ace your initial verbal interview, when it comes to hiring for any software development roles, the core of the whole process is for hiring managers to determine your ability to write quality code which is correct, efficient and at industry standards. Often they achieve this through White Board technical questions and take home tests/assessments - in this task we will focus on typical take home tests.

The main aim of these kind of assessments goes beyond simply testing your ability to solve the problem. There are many other aspects which hiring managers look out for and aim to learn about you. These include but are not limited to:

- What is your overall approach to solving a problem?
- What do you do when you get stuck?
- Are you able to 'brute force' your way to the solution?
- Are you methodical or do you just hammer through a bunch of approaches and see what works?
- Do you leave any documentation and comments on your code?

Often, interviewers are not even looking for the right answer - they're trying to test your sense of logic, your ability to handle pressure and think clearly in a high-pressure situation, and your ability to think through problems while asking questions.

So, your best bet is to take a moment to organise your thoughts, and then think through the question verbally. Ask clarifying questions. Logically analyze (out loud) the different ways to approach it. Discuss why you would prefer approaching the problem in a certain way, and - if you get there - eventually describe your answer.

Remember: getting the right answer (or one of the right answers) is important, but equally important is to showcase yourself as a logical, verbal, and analytical person who can hold on to their wits even in a stressful situation.

Here are some resources which will help you prepare for the programming part of technical interviews:

- How to Crack Coding Interviews
- Full Stack Developers Interview questions and answers
- Software Engineers Interview questions and answers
- General Mobile Developer Interview questions and answers
- Android Developers Interview questions and answers

In this task, we will cover a few key concepts and algorithms that candidates are most frequently tested on.

In a similar manner to all your previous tasks, please attempt these tasks in your Dropbox folder and update your comments.txt file to inform your mentor when done!

Compulsory Task 1

Dynamic Programming

Check if any valid sequence is divisible by M

Given an array of N integers, using '+' and '-' between the elements check if there is a way to form a sequence of numbers which evaluate to a number divisible by M

Examples:
Input : arr = {1, 2, 3, 4, 6}
M = 4
Output : True,
There is a valid sequence i. e., (1 - 2 + 3 + 4 + 6), which evaluates to 12 that is divisible by 4

Input : arr = {1, 3, 9}
M = 2
Output : False
There is no sequence which evaluates to a number divisible by M.

Hint: A simple solution is to recursively consider all possible scenarios, i.e either use a '+' or a '-' operator between the elements and maintain a **variable sum** which stores the result. If this result is divisible by M, then return true else return false.

Compulsory Task 2

Arrays & String manipulation

Given a string, calculate and print all its possible palindromic partitions. *(A palindrome is* a word, phrase, or sequence that reads the same backwards as forwards)

Examples:
Input: nitin
Output: n i t i n
n iti n
nitin

Input: geeks
Output: g e e k s
g ee k s

Hint: Go through every substring of the given string starting from first character, check if it is a palindrome. If yes, then add the substring to the solution/display and traverse through remaining part. The use of recursion is handy for this kind of a problem!

The next step is a mock interview with the HyperionDev Career Support Team. Please get in touch with us at support@hyperiondev.com to let us know if you would like to partake in a mock interview and we will set up one accordingly upon successful completion of this task. :)

We are excited to have walked this coding journey with you and we look forward to working with you further as you start your new career in tech!

SHARE YOUR THOUGHTS WITH US

Hyperion strives to provide high quality and internationally relevant course content that helps you achieve your learning outcomes. Think the content of this task (or this course as a whole) can be improved, or think we've done a good job? Click here to share your thoughts anonymously.