verticalmove™

Assessment Profile™

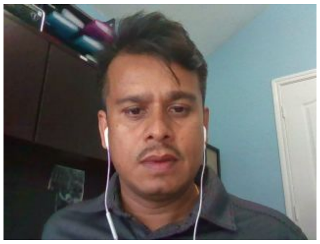
Candidate ID: **7194216**

Candidate Name: **Babu**

Date: **27-Sep-2022**

Assessment Name: **Backend NodeJS & API**

Client Name: **verticalmove.com**

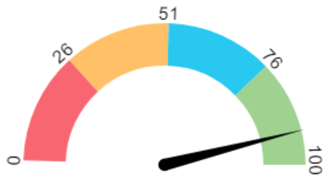


SCORE ANALYSIS

Score: **14/ 15**

Time Taken:
27 min 50 sec / 35 min

Proficient (93%)



🟥 BEGINNER (0% - 25%)

🟧 INTERMEDIATE (26% - 50%)

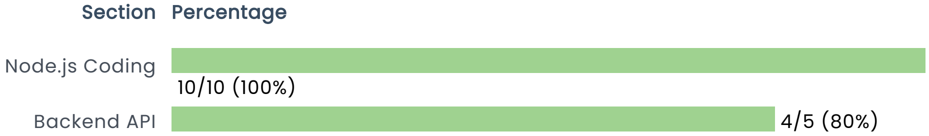
🟦 EXPERIENCED (51% - 75%)

🟩 PROFICIENT (76% - 100%)

Babu scored **93%** and completed assessment in **80%** of the alloted time

SECTION SCORE ANALYSIS

🟥 BEGINNER (0% - 25%) 🟧 INTERMEDIATE (26% - 50%) 🟦 EXPERIENCED (51% - 75%) 🟩 PROFICIENT (76% - 100%)



SECTION SKILL ANALYSIS

Section 1: Node.js Coding

Total Score: **10/ 10** Negative Points: **0** Time Taken: **25 min /25 min**

Question Analysis:

Total Question: **1** Correct: **1** Wrong: **0** Skipped: **0** Not Answered: **0**

Skills	#Questions	Skill Score
Node.js Coding	1	10/10

Section 2: Backend API

Total Score: **4/ 5** Negative Points: **0** Time Taken: **2 min 50 sec/10 min**

Question Analysis:

Total Question: **5** Correct: **4** Wrong: **1** Skipped: **0** Not Answered: **0**

Skills	#Questions	Skill Score
Backend API	5	4/5

DIFFICULTY LEVEL ANALYSIS

Level	Number of Questions	Correct Attempts	Correctness
Easy	0	0	0%
Medium	6	5	83.33%
Hard	0	0	0%

PROCTORING ANALYSIS

Images Captured: **58**	Image Violations: **4**

Image violations detected, within tolerable limit.

Window Violation: **0**	Time Violation: **0 sec**

QUESTION DETAILS

Question: **#1**	Type: **Coding**	Skill: **Node.js Coding**	Status: **Answered**
Result: **Correct**	Level: **Medium**	Time Taken: **25 min 0 sec**	Average Time: **15 min 11 sec**
Score: **10 / 10**	Window Violation: **0 times**	Time Violation: **0 sec**	

Question #1

Algorithms: Load Distribution

You are working as a server maintenance engineer at an IT company. The company uses 2 server architectures, the main server, and the support server to execute processes.

You try to execute most of the processes on the main server, at the same time you don't want to execute heavy processes on the main server as it will increase the load on the main server and will decrease the performance of the main server. To handle this problem you decide to route the heavy processes to the support server.

You are given an integer array that represents the time (in seconds) required to execute a pool of independent processes.

Your task is to identify the minimum number of processes that need to be routed to the support server so that the load on the main server is less than that of the support server. After the routing, there should not be any process for execution on the main server whose execution time is more than the execution time of the minimum execution time process on the support server. A load on a server is defined as the sum of time required by different independent processes for execution.

Function Description

In the provided code snippet, implement the provided `loadDistribution(...)` method using the variables to print the number of units of gift X and Y you need to make to maximize your sales. You can write your code in the space below the phrase “WRITE YOUR LOGIC HERE”.

There will be multiple test cases running so the Input and Output should match exactly as provided.

The base Output variable result is set to a default value of `-404` which can be modified. Additionally, you can add or remove these output variables.

Input Format

An integer array in which the ith integer represents the time required to execute the ith process in a pool of independent processes.

Sample Input

```
1 2 3 4 1 3 1 8
```

Output Format

A space-separated integer array representing the time required to execute the processes that is/are being routed to the support server in descending order.

Sample Output

```
8 4
```

Explanation

After Routing the load on
Main server = 1 + 2 + 3 + 1 + 3 + 1 = 11
Support server = 8 + 4 = 12.
Hence the output is ***8 4***.

Answer:

Coding Language: Node.js

Candidate Code:

```
function loadDistribution(seconds) {
  let totalTime = 0;
  seconds.sort((a,b) => a-b);
//this is default OUTPUT. You can change it.
for (let i =0; i<seconds.length; i++){
  totalTime += parseInt(seconds[i]);
}
let response = [];
let currentTotal = 0;
for (let i=seconds.length-1; i>=0; i--) {
  currentTotal += parseInt(seconds[i]);
  response.push(seconds[i]);
  if (totalTime / currentTotal >= 2) {
    continue;
  } else {
    break;
  }
}

var result=response.join(" ");

//write your Logic here:
```

```
return result;
}

// INPUT [uncomment & modify if required]
var temp = gets().trim('\n').split('\n');
var seconds = temp[0].split(" ");

// OUTPUT [uncomment & modify if required]
console.log(loadDistribution(seconds));
```

Compilation Summary:

Compilation Status: **Compile Successfully**
Defualt Input:
1 2 5 7 8 11 2 13 9 7 5 6

No Of Compilations: **14**
Candidate Output:

```
13 11 9 8
```

Test Case Summary:

Test Case: **1** Status: **Pass** Score:**0**

Test Case Input	Expected Output	Actual Output
1 2 3 4 1 3 1 8	8 4	8 4

Test Case: **2** Status: **Pass** Score:**2**

Test Case Input	Expected Output	Actual Output
1 2 1 1 1 6	6 2	6 2

Test Case: **3** Status: **Pass** Score:**2**

Test Case Input	Expected Output	Actual Output
1 2 5 7 8 11 2 13 9 7 5 6	13 11 9 8	13 11 9 8

Test Case: **4** Status: **Pass** Score:**2**

Test Case Input	Expected Output	Actual Output
17 9 13 11 8 9 5 17 2 10 11 16 3 9 6 15 1 4 11 12 10 11 6 8 3 1	17 17 16 15 14 13 12 11 11	17 17 16 15 14 13 12 11 11

Test Case: **5** Status: **Pass** Score:**2**

Test Case Input	Expected Output	Actual Output

Test Case Input	Expected Output	Actual Output

4
6 7 4
2 2
4 3
0 4
1 6 1
0 3
2 2
8 13 1
8 1
8 4
7 12 2
7 3
9 2
4 4
7 1
0 2
9 5
0 1
5 4
5 4
1 3
8 4
3 4
7 4
9 1
7 3
9 1 3
3 2
8 16 1
7 18 1
4 6 7 4
2 4
2 4
8 5
0 7 2
4 5
0 4
5 3
2 43
	50 50 50 49 48 47 47 47 46 45 45 43 43 42 42 42	50 50 50 49 48 47 47 47 46 45 45 43 43 42 42 42

Test Case: 6 Status: Pass Score:2

Test Case Input	Expected Output	Actual Output
8 8 1 1 9 1 2 8 7 4 1 3 8 9 8 4 3 2 1 6 8 7 4 4 9 9 1 9 1 7 8 1 2 5 2 5 3 4 9 5 2 5 3 2 9 3 9 2 2 2	9 9 9 9 9 9 9 9 8 8 8 8 8 8	9 9 9 9 9 9 9 9

How We Verify Candidate Identity

Verticalmove, Inc has a thorough, ongoing process to validate identity and guard against candidates attempting to cheat during any part of the vetting and interview process - all of which are designed to create the best possible delivery of resources, candidate experience, and client experience. This is a cooperative effort that requires some participation on your end, to close the circle of doubt.

verticalmove™

Verticalmove Process

01

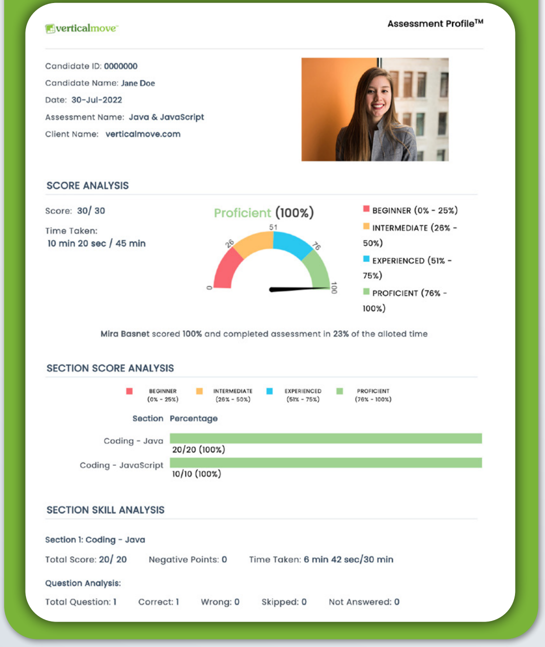
Every candidate receives a technical assessment to validate and compare candidate's technical skillsets and abilities - The assessment test has 2 primary security tools for catching cheaters:

A The system takes their picture every 30 seconds

i. If there is a second face in the frame, or if a candidate is looking offscreen (presumably looking at their phone or another monitor for answers), or if there are any image inconsistencies at all – they receive 4 warnings and on the 5th offense the test auto-ends in failure

B If the candidate tries to click on a different window (presumably to screen share or lookup answers) – they receive 1 warning, and on the 2nd offense the test auto-ends in failure

Technical Assesment Profile Example



verticalmove

Assessment Profile™

Candidate ID: 0000000
Candidate Name: Jane Doe
Date: 30-Jul-2022
Assessment Name: Java & JavaScript
Client Name: verticalmove.com

SCORE ANALYSIS

Score: 30/ 30

Time Taken:
10 min 20 sec / 45 min

Proficient (100%)

- 🟥 BEGINNER (0% - 25%)
- 🟧 INTERMEDIATE (26% - 50%)
- 🟦 EXPERIENCED (51% - 75%)
- 🟩 PROFICIENT (76% - 100%)

Mira Basnet scored **100%** and completed assessment in **23%** of the alloted time

SECTION SCORE ANALYSIS

🟥 BEGINNER (0% - 25%) 🟧 INTERMEDIATE (26% - 50%) 🟦 EXPERIENCED (51% - 75%) 🟩 PROFICIENT (76% - 100%)

Section	Percentage
Coding - Java	20/20 (100%)
Coding - JavaScript	10/10 (100%)

SECTION SKILL ANALYSIS

Section 1: Coding - Java

Total Score: 20/ 20 Negative Points: 0 Time Taken: 6 min 42 sec/30 min

Question Analysis:

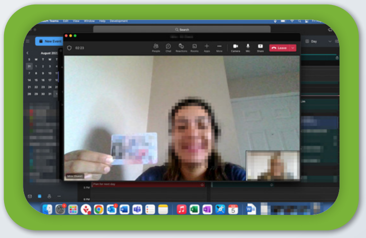
Total Question: 1 Correct: 1 Wrong: 0 Skipped: 0 Not Answered: 0

02

Only candidates in the top 25 percentile of test-takers move on to the next round

03

During the internal interview & vetting process, our recruiters take a screenshot of the candidate holding their Identification card. This validates the person on the call is the same person on the ID, which is the same person who took the test



04

All images are sent to you in the submittal

What We Need from You

01

Most Importantly, above all else – Video interviews only. No interviews should ever be conducted via a phone call. Every interview must require the candidate to be on a live video call, with a good connection.

02

We request to be added to the video interview invite so we can login to verify candidate ID, on your behalf, before the call starts. We will either validate quickly or end the call immediately.

03

Before the interview, you will have 3 different pictures we've sent you – The ID, a screenshot of their face, and the picture from the assessment test

a. You need to have those pictures up during the interview to confirm that the person being interviewed is the same person that we sent you.

04

This article explains some of the incredibly manipulative methods that deepfakes use to cheat

a. If, at any time, the candidate appears to be lip-syncing, over-compensating communication during the interview, or shows irregular behavior - Please cancel the interview and let us know immediately. Do not go on with the interview.

b. If the candidate starts to experience sudden "connectivity issues" that are not quickly fixed – end the interview. It is likely that they are a deepfake – or at the very least, they're really having connection issues, in which case you wouldn't want them on your team because they would likely continue to experience those issues while on the job.

verticalmove™

What We Need from You

05

Interview Questions: When interviewing a candidate on technical knowledge, ask problem-solving questions that cannot be looked up on Google.

a. Do not ask questions that can be looked up on Google to find the answer. This will ensure you are receiving real answers directly from the candidate.

b. If the candidate clearly and obviously cannot answer your initial, basic technical questions, end the call immediately and notify your Verticalmove rep.

06

Day 1 On the Job Virtual Video Call: This is possibly the most important part of the process because we need to verify that the person you interviewed is the same person that shows up for work. Verticalmove requires a "pass-off" Virtual Video call on day 1 of employment so we can introduce the newly onboarded resource to your team. This call will include the candidate, the candidates reporting manager, and your Verticalmove rep. We will require them to hold up their ID so we can match the person and their ID with the pictures & ID we sent you in the original submittal.

07

Require the resource to post a profile picture on whatever internal communication service you use (Slack, Zoom, TEAMS, WebEx, etc). This should be a requirement on your end before they can begin any type of work.

a. Always require the resource to turn their camera on when speaking with other team members or leaders. This creates a culture of accountability and the expectation that their image will continuously be monitored.

verticalmove™