Technical Problem Review

The first part of your Engineering interview includes producing some sample code for you to review with the interviewer during the Technical Problem Review interview. Please choose from any of the problems outlined below and code your solution in the language of your choice.

Prior to starting your project, your recruiter will schedule a 15-minute Q&A session via Zoom with the interviewer so that you have the opportunity to get clarification on the problems outlined and ask any questions you may have. After your 15-minute Q&A, you will begin working on the project, of your choice, over the next few days. The 1-hour Problem Review and code review will be scheduled via Zoom a few days after your Q&A session. Be sure that you have provided your GitHub username to your recruiter. You will receive an invite to a code repository on the HEB-Recruiting GitHub account at https://github.com/heb-recruiting where you will submit your code for review. Please note you are not expected to spend more than 2-hours of your time coding your solution.

How to prepare:
- Ask questions during both the Q&A and the Problem Review! The interviewer is not here to trick you, they are here to help you and are trying to get a sense of what it would be like working with you on a problem collaboratively.
- Make sure to review your code and are able to walk someone through your decisions.
- If you use any 3rd party libraries, make sure you know general details of their underlying implementation and complexities.
- The reviewer may ask you about test cases you haven't considered, or introduce new complexities not included originally, think out loud as you work through how your code may be adjusted to handle them.
- Have your code open in your IDE so you can share your screen.
- Come to the Problem Review prepared to discuss the algorithms, data structures, design patterns, and the scalability of the solution presented.

Problem Statements

Select one (1) of the problems outlined below to work a solution for in the coding language of your choice.

Shopping Cart Problem

We sell a lot of groceries, so we fill and empty tons of virtual shopping carts. To sell groceries online we need a few features the business has asked for. Below are four shopping cart features the business has requested.

This will be a multi-part question and the expectation is you likely won't complete it in the 2-hour time allotment. Be sure that you have provided your GitHub username to your recruiter. You will receive an invite to a code repository on the HEB-Recruiting GitHub account at https://github.com/heb-recruiting where you will submit your code for review.

If you find any issues with the files we have sent or have other questions don't hesitate to reach out to your recruiter or the H-E-B Partner that you met with during the 15-min Q&A session.

Assumptions

- All items are single quantities
- The shopping list is static for each feature section
- The coupon list is a separate JSON file
- In feature 4, the final price of an item cannot be negative
- The tax rate is .0825(8.25%)
- All prices are in USD

Feature 1

Calculate the total of the given shopping cart. Print out the Grand Total of the shopping cart.

Feature 2

Calculate the total of a given shopping cart and include the sales tax amount from the subtotal of the cart. Print out the following:

- Subtotal
- Tax Total
- Grand Total

Feature 3

Not all are items are taxable. Re-calculate the tax total based on the boolean field in the JSON file "isTaxable". If "isTaxable" is true then calculate sales tax for that item, if it is false skip the sales tax calculation. Print out the following:

- Subtotal
- Tax Total
- Grand Total

Feature 4

The business would like to support coupons for certain items in a cart. Apply the coupon to a customer's shopping cart if they have any of the items in their cart associated with coupons contained in the coupon.json file.

Price Comparison Tool Problem

This will be a multi-part question and the expectation is you likely won't complete it in the 2-hour time allotment. Be sure that you have provided your GitHub username to your recruiter. You will receive an invite to a code repository on the HEB-Recruiting GitHub account at https://github.com/heb-recruiting where you will submit your code for review.

If you find any issues with the files we have sent or have other questions don't hesitate to reach out to your recruiter or the H-E-B Partner that you met with during the 15-min Q&A session.

Build a price comparison tool that will get pricing and availability data from these sources:

URL	appedia.fake/api/v1/itemdata?upc=12345

Response	`{`
	`  "price": "$5.67",`
	`  "stock": 7`
	`}`

Description	Price is returned as a string Availability('stock') returned as an integer value, 0 meaning out of stock.

URL	micromazon.lol/12345/productinfo
| Response | ```json
{
  "price": 4.56,
  "available": true
}
``` |
| Description | Price is returned as a double

Availability('available') returned as a boolean value |

URL	googdit.nop/12345
| Response | ```json
{
  "p": 234000000,
  "a": [
    {
      "l": 123,
      "q": 0
    },
    {
      "l": 234,
      "q": 5
    }
  ]
}
``` |
| Description | Price is returned as microcents (ie 234000000 == $2.34)

Availability('a') is returned as an array of objects with quantity('q') available at location('l') |

Requirements

- Return the URL that has the lowest price and has the item in stock at any location.
- Make it easy to swap item UPC.
- Make it easy to add additional merchant URLs to query for price comparison.
- Write unit/integration tests.

Hypervisor Problem

We manage over 10,000 virtual servers and over 500 applications to run our business in on-prem data centers and edge data centers. After performing a weekly system audit, we found several systems in one cluster that need specific features to have them balanced and remediated to ensure optimum performance.

This will be a multi-part question and the expectation is you likely won't complete it in the 2-hour time allotment. Be sure that you have provided your GitHub username to your recruiter. You will receive an invite to a code repository on the HEB-Recruiting GitHub account at https://github.com/heb-recruiting where you will submit your code for review.

If you find any issues with the files we have sent or have other questions don't hesitate to reach out to your recruiter or the H-E-B Partner that you met with during the 15-min Q&A session.

Assumptions

- List of VMs provided (in VM_Info.json) are all VMs in a cluster that host several different applications.
- Naming schema for VMs are as follows:

    - [W2] [C] [AP] [L] [00] [777] [98] - Server Name - w2capl0077798
    - [ZONE] [Environment] [Type of Server (Application , Database etc)] [Base O/S (Linux or Windows)]
    - [Reserved Numbers] [Application Group] [Server Number]
- Cluster size is 8 hosts, with 48 cores and 2 sockets each, and 768 GB memory per host. All storage is SAN based block storage for all VMs, sizing is 2 GB storage per VM (not articulated in the VM_Info.json).

Feature 1

Some of the servers in the cluster are not sized correctly. Based on the list of VMs provided, for all "POWERED_ON" VMs, supply the output of the list of VMs that do not match standard memory to vcpu NUMA configuration for optimized performance. Based on this list, write a function that will resize these VMs to the appropriate configuration for optimized performance.

Feature 2

VMs in the 777 authorized server group are showing performance issues. For all "POWERED_ON" VMs in the 777 authorized server group, write a function to double the memory and CPU for each of these VMs. If they are in a non-standard config, ensure that the

configuration is set correctly as per optimization standards, then double the size of the CPU and memory for each of the 777 authorized server group VMs.

Feature 3

You have written the code for resizing, but want to ensure that you have a logic check to ensure there is enough resource space on the cluster before executing the commands for any cpu / memory modifications. Write the function that will check the host and cluster resources to ensure enough space is available to perform the functions. Ensure that the resource increase does not take up more than 75% of usable space on any 1 host.

Feature 4

Write the function that will remove all "POWERED_OFF" VMs from inventory.

Basic Scenario 1

You need to build out a new Active / Active Cluster across two data centers. Build out what would be the minimum configuration for hosts per Data Center, including your interpretation of sizing for compute and storage, HA, and DR ability. How would you design this?

Requirements:

- Can house 25 VMs at a 4 VCPU and 16 GB Memory Footprint, each with 1 TB Memory
- Must be able to run out of both site simultaneously, or out of a single site in case of a disaster scenario
- Must have all data backed up
- Must be able to be scaled for future growth

Basic Scenario 2

You have a set of VMs running in a cluster that cannot reside on the same host due to communication constraints. How would you architect a set of 5 VMs on a 4 Node cluster where each cannot reside on the same host simultaneously.