# Read First

You have **24 hours** to resolve each of the following questions we are setting. You have time to resolve as much as you can. You have to resolve every question with coding because you are going to be a programmer.

Please answer every question using tech stacks you are applied to.

1. If you are done with your task. You must push them all in the public git repository. Must **push,** **not** **upload.** Pushing code by following git flow standard.
2. Also with the conventions, every answer you made, each must have a standalone folder with name Q1, Q2, Q3, Q4, Q5 and push them in one repository
3. Don’t ever push them in compressed format like .zip, .rar. .gzip, etc.
4. Every folder of the answer must be a runnable/executable application when we `cd` to the directory.
5. Every folder of the answer must have a README.md explaining what it is.

You are The Best, Good luck!

# 1. Story

A small restaurant wants to have an app to print a payment receipt, the resto has a dot matrix printer. While you developing the app to print that receipt, you have to make sure every input and output working as expected when printing the receipt by following the required parameters of dot matrix printer.

**Note: Don’t forget give description in the README.md**

**Given**:

1. Max width of one line is 30 characters.

**Known/Constraints**:

1. When text is more than 30 characters, the rest must be over to the new line.
2. Output at the section of price list must be connected with dot character (...) between the item name until the item price.
3. Item name always **aligned to left**, Item price always **aligned to right with Rp as prefix**.

**Expected Input**:  
Simple app, you can just make it running on command line, no fancy here, just must have there inputs:

1. Input resto name
2. Input date of print
3. Input cashier name
4. Input item and price, for this can be multiple times input until the user said exit.

**Expected Output**:

Warung Makan Sederhana

Tanggal : 2020/12/2020 15:30:00

Nama Kasir : {{YourName}}

================================

Nasi...................Rp5.000

Lauk...................Rp10.000

Minum...................Rp5.000

Total...................Rp20.000

# 

# 2. JSON Manipulation

We have JSON Data:

<https://gist.github.com/dhamanutd/6993984928506eea49908c2e3fcbc628>

Your tasks to create functions:

1. Find users who don't have any phone numbers.
2. Find users who have articles.
3. Find users who have "annis" on their name.
4. Find users who have articles on the year 2020.
5. Find users who are born in 1986.
6. Find articles that contain "tips" on the title.
7. Find articles published before August 2019.

# 

# 3. JSON Manipulation

We have JSON Data:  
<https://gist.github.com/dhamanutd/97aa0d2131903ea8c071721032c7b2a3>

Your tasks to create functions:

1. Find items in the Meeting Room.
2. Find all electronic devices.
3. Find all the furniture.
4. Find all items were purchased on 16 Januari 2020.
5. Find all items with brown color.

# 4. Debugging

We want to know good you are in problem solving, in this section you will handle basic errors and most common development issue. Because programmers are problem solvers that use programming languages as their tools.

We don’t want to see how expert you are at some languages, we want to know how good you are to adapt every challenge, then find the problems, and also the solutions.

**Choose at least 1 stack you want to resolve on these below options, write down your found issues, analytics, and also the solutions in the README.md, and also don’t forget to fix the code too, not just the README.md.**

**If you can solve more than one tech stacks, it will be better for you.**

## **FE & Mobile**

Clone and run one of these repo to your local machine. Those apps have some errors, find out and fix them. If you are done, please push them to your repository.

### **FE**

* VueJS: <https://github.com/refactory-id/recruitment-vuejs-test>
* ReactJS: <https://github.com/refactory-id/recruitment-reactjs-test>

### **Mobile**

* iOS Swift: <https://github.com/refactory-id/recruitment-ios-test>
* Android Kotlin: <https://github.com/refactory-id/recruitment-android-test>
* Flutter: <https://github.com/refactory-id/recruitment-flutter-test>

## **Backend**

1. **NodeJS**: <https://github.com/refactory-id/recruitment-code-debugging>.  
   Expected return response:

{

data: {

login: 'mazsam',

githubId: 17076855,

avatar: 'https://avatars2.githubusercontent.com/u/17076855?v=4',

email: 'arifinsam.sam@gmail.com',

name: 'Samsul Arifin',

location: 'Blitar, Indonesia'

}

}

1. **Golang**Please open this link to solve the problem <https://gist.github.com/aditiapratama1231/473ca56d498843c733deb3f7075e6606>

# 5. Simple APP

## **FE/Mobile**

* Please create UI that following this design and flow in this figma, **and you can improvise to impress us**:  
  <https://www.figma.com/file/uvLIMVVlssrWOYa8xMfvnQ/Untitled?node-id=0%3A1>

1. Contains 1 screen
2. Have calendar
3. Have modal

## **BE**

Build client-server app regarding these criterias:

1. Every minutes client will send POST request to server with following payload::

// Header

"X-RANDOM": "93f9h3dx"

// Body

{ "counter": 1 }

// Header

"X-RANDOM": "fe9g83jm"

// Body

{ "counter": 2 }

// Header

"X-RANDOM": "igrijd9p"

// Body

{ "counter": 3 }

1. Server will receive a request from the client above. Then the server will save those received request to **server.log** file. After that server must return a response to the client with HTTP status code 201. Example server.log:

[2020-07-28T16:23:40+07:00] Success: POST http://192.168.1.30/ {"counter": 1, "X-RANDOM": "93f9h3dx"}

[2020-07-28T16:24:40+07:00] Success: POST http://192.168.1.30/ {"counter": 2, "X-RANDOM": "fe9g83jm"}

[2020-07-28T16:25:40+07:00] Success: POST http://192.168.1.30/ {"counter": 3, "X-RANDOM": "igrijd9p"}