

# Recruitment task - Dec 2022

## Codeware Limited

### Tech stack

#### Backend

- **NodeJS**
- **MongoDB**

\*\*deploying to a cloud service like heroku, render or aws is a plus

#### Frontend

- **ReactJS**

\*\* You can use a CSS library like styled-components or JSS

\*\* Deploying to a cloud service like netlify or vercel is a plus.

#### Note:

\*\* Using **typescript** is highly preferred for both backend and front end.

\*\* You must submit a link to your work - use a git cloud service like Bitbucket, Github or Gitlab.

\*\* Submit links if you deploy your work to a cloud service

### 1. Age counting

Stack : javascript

GET request to find those keys as array where age's value is equal to 30 from the API below

<https://coderbyte.com/api/challenges/json/age-counting>

### 2. Countdown Sequence

Stack : javascript

A countdown sequence is a descending sequence of k integers from k down to 1 (e.g. 5, 4, 3, 2, 1). Write a function that returns an array of the form [x, y] where x is the number of countdown sequences in the given array and y is the array of those sequences in the order in which they appear in the array.

### Example inputs and outputs:

```
finalCountdown([4, 8, 3, 2, 1, 2]) → [1, [[3, 2, 1]]]  
// 1 countdown sequence: 3, 2, 1
```

```
finalCountdown([4, 4, 5, 4, 3, 2, 1, 8, 3, 2, 1]) → [2, [[5, 4, 3, 2, 1], [3, 2, 1]]]  
// 2 countdown sequences:  
// 5, 4, 3, 2, 1 and 3, 2, 1
```

```
finalCountdown([4, 3, 2, 1, 3, 2, 1, 1]) → [3, [[4, 3, 2, 1], [3, 2, 1], [1]]]  
// 3 countdown sequences:  
// 4, 3, 2, 1 ; 3, 2, 1 and 1
```

```
finalCountdown([1, 1, 2, 1]) → [3, [[1], [1], [2, 1]]]
```

```
finalCountdown([]) → [0, []]
```

### 3. Jake's Meal Time

Stack : javascript

Jake is a very habitual person. He eats breakfast at 7:00 a.m. each morning, lunch at 12:00 p.m. and dinner at 7:00 p.m. in the evening.

Create a function that takes in the current time as a string and determines the duration of time before Jake's next meal. Represent this as an array with the first and second elements representing hours and minutes, respectively.

### Example inputs and outputs:

```
timeToEat("2:00 p.m.") → [5, 0]  
// 5 hours until the next meal, dinner
```

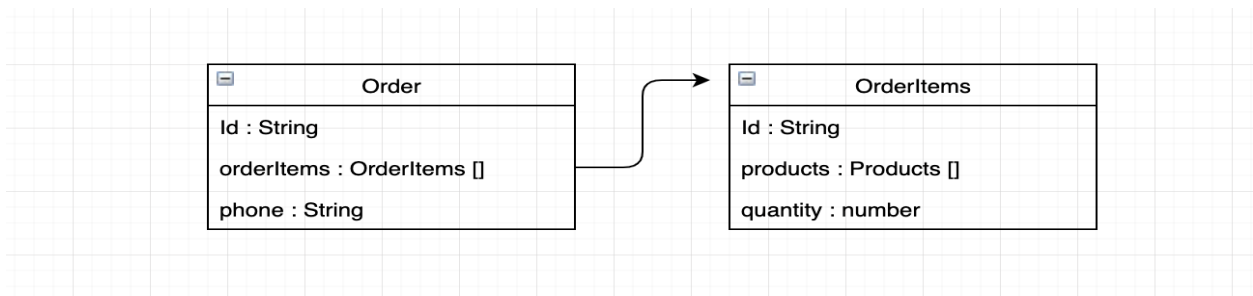
```
timeToEat("5:50 a.m.") → [1, 10]  
// 1 hour and 10 minutes until the next meal, breakfast
```

## 4. Placing Order

Stack : Nodejs / expressjs, MongoDB

Here we have two models . First one is Order and the second one is OrderItems. As you can see there is a relation between them.

When the user select products and creates an order we have to save the **Order** and **OrderItems** in different collections in mongoDB .



Things we expect :

Post request to make order(along with orderitems in different collections same as picture).

Order response(get) :

/\*\*

Order Example:

```
{
  "orderItems": [
    {
      "quantity": 3,
      "product": "Orange"
    },
    {
      "quantity": 2,
      "product": "Banana"
    }
  ],
  "phone": "+420702241333",
}
```

\*\*\*you don't need to make product table , just pass string in orderItems (for example : Banana , Orange)

## 5. Simple Button Click Event

Stack : Reactjs

Write a React component that presents users with a button. The button text says "click me" but on every third click (e.g. the 3rd click, the 6th click, the 9th click, etc...), the button text changes to "bang" but reverts back to "click me" when a users presses it again. Remember to properly export your component using export default.