Full Stack Developer Challenge

This challenge is designed to assess your technical skills to determine if it aligns with our requirements.

Duration

1-2 days

Important

The test is designed to be performed in the above-mentioned time. In case you cannot complete it in its entirety, please make sure you fill in all the mandatory fields (those with * symbol) and that you provide some placeholders for the missing parts.

Terms of Delivery

Please provide a GIT-HUB link for both exercises

Don't worry if you're unable to complete everything. Focus on code quality and let us know about any parts you couldn't finish due to time constraints. Please include an explanation of how you would have implemented those sections.

elastik

Code Challenge*

Loraine and Charles have a nearly 3-year-old child! Although their child is more independent now, scheduling activities and household tasks **remains** challenging for them.

They have a list of **N daily activities**. Each activity **occurs**within a specified time interval. They must assign each activity to one of them, ensuring that neither person is responsible for two overlapping activities. **An activity ending at time t does not overlap with another starting at time t.**

For example, suppose Loraine and Charles need to cover three activities: one from 18:00 to 20:00, another from 19:00 to 21:00, and another from 22:00 to 23:00. One option is for Charles to handle the activity from 19:00 to 21:00 while Loraine covers the other two. Another valid schedule could have Loraine covering the activity from 18:00 to 20:00, with Charles covering the other two. **Note** that the first two activities overlap between 19:00 and 20:00, so they cannot be assigned to the same person.

Given the start and end times of each activity, find any schedule where the same person doesn't handle overlapping activities, or state that it is impossible.

Input

The first line of input specifies the number of test cases, **T**.

Each test case begins with a line containing a single integer **N**, indicating the number of activities to be assigned.

Then, **N lines follow**. The **i-th** line (starting from 1) contains two integers **Si** and **Ei**.

The i-th activity starts **exactly Si** minutes after midnight and ends **exactly Ei** minutes after midnight.

Output

For each test case, output one line containing Case #x: y, where **x** is the test case number (starting from 1) and **y** is IMPOSSIBLE if there is no valid schedule according to the above rules, or a string of exactly N characters otherwise. The i-th character in y must be C if the i-th activity is assigned to Loraine in your proposed schedule, and J if it is assigned to Charles.

Sample

elastik

Input	Output
4 3 360 480 420 540 600 660 3 0 1440 1 3 2 4 5 99 150 1 100 100 301 2 5 150 250 2 0 720 720 1440	Case #1: CJC Case #2: IMPOSSIBLE Case #3: JCCJJ Case #4: CC

Sample Case #1 is the one described in the problem statement. There are other valid solutions, like `JCJ` and `JCC`.

In Sample Case #2, all three activities overlap with each other. Assigning them all would mean someone would end up with at least two overlapping activities, so there is no valid schedule.

In Sample Case #3, notice that Loraine ends an activity and starts another one at minute 100.

App Challenge

Elastik is an innovative platform that supports, connects and empowers everyone involved in a child's school journey. Pupils, teachers, parents and principals, all contributing and collaborating to help children fulfil their potential.

For this exercise, you are about to implement the required tasks below:

1. Create a basic login page for teachers.*
2. Create a basic module, fully functional, for teachers to create students, this will include student name, last name, email, DOB and ID, validations are up to you but at least ID and first name must be filled.*
3. Create a module to display the students created by the teacher.*
4. A teacher should be able to remove students.

elastik

Notes:

1. There are not suggested designs, you can propose your own UI/UX
2. The frontend technology to use is ReactJs, you are free to choose any UI library, we recommend CoreUI or DevExtreme*
3. The backend should be developed with AWS cloud services such as Amplify, Cognito, Lambda functions, Dynamo, Appsync.
4. Please implement the login process using a Lambda function connected to an API Gateway, which then calls Cognito operations through the AWS SDK.
5. Try to include the use of git in the development process. For example, you may want to create a branch main and then create 4 PRs to that branch for tasks 1-4.*
6. In the end, please provide a GitHub link with the project you developed, include any instructions to set it up in case it will be necessary. *

Bonus tasks:

Feel free to also implement any of the following tasks if you are just getting warmed up:

1. Add Accessibility (A11y) testing and update components to comply.
2. Create a dark theme of the project.
3. Implement a logged user widget, displayed in the upper-right corner.
4. Make the app responsive at least on tablets.
5. Write some tests(Jest, Cypress, React Testing Library)
6. Add documentation about your tasks to help other developers understand decisions you made.