Prompts

1. **Fixing Attendance Marks Calculation:**
    - While incrementing attendance marks, if the category is `"cr"`, ensure that the function skips it using a `continue` statement. This will prevent unnecessary calculations for `"cr"` categories.
2. **Converting JSON to Excel:**
    - How can I convert the generated student data (JSON) into an Excel sheet?
    - The Excel sheet should contain:
        - Student Name
        - Enrollment Number
        - Each subject's details, including:
            - Subject Name
            - Marks for each individual category
            - Grades for each individual category
            - Total grade
        - SPI (Semester Performance Index)
    - The `"cr"` category should be excluded from the Excel output.
3. **Unit Testing:**
    - Now that the application is working, how do I write unit test cases for it?
    - Specifically, I need test cases for:
        - `addMarks`
        - `calculateAttendanceMarks`
        - `calculateSPI`
        - `isStudentPassInAllSubjects`
        - `parseExcel`
        - `writeStudents`
        - `writeStudentsToExcel`
4. **Test Case for** `isStudentPassInAllSubjects`**:**
    - For testing `isStudentPassInAllSubjects`, use the following input:

```
json
CopyEdit
{
  "Enrollment": 21012250210001,
  "Name": "AARSH VACHHANI",
  "Att": 7,
  "Subjects": {
    "PROJECT II": { "PR": { "Score": 90, "Max": 100 } },
    "MCAD": {
      "cr": { "Score": 5, "Max": null },
      "TH": { "Score": 9, "Max": 50 },
      "PR": { "Score": 38, "Max": 50 }
    },
    "IOT": {
      "cr": { "Score": 4, "Max": null },
      "TH": { "Score": 39, "Max": 50 },
      "PR": { "Score": 45, "Max": 50 }
    },
    "ML": {
      "cr": { "Score": 5, "Max": null },
      "TH": { "Score": 30, "Max": 50 },
      "PR": { "Score": 44, "Max": 50 }
    },
    "Seminar": {
      "cr": { "Score": 2, "Max": null },
      "PR": { "Score": 90, "Max": 100 }
    }
  }
}
```

- This should cover a valid test case where all subjects are passed. Also, write another test case where at least one subject does not meet the passing criteria.

5. **Handling Module Import Issues in Jest:**

- When using `require`, Jest throws the following error:

```
pgsql
CopyEdit
Cannot find name 'require'. Do you need to install type definitions for node? Try
`npm i --save-dev @types/node`.
```

How can I resolve this?

- If I switch to `import`, Jest throws:

```
javascript
CopyEdit
SyntaxError: Cannot use import statement outside a module
```

What is the correct way to import functions from an `.mjs` file in Jest?

6. **Finalizing the Testing Setup:**

- Now that the import issue is resolved, can you provide test cases similar to the one for `addMarks`, but for the other functions as well?