Question 1:

Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <th>, <td>, and <thead>.

→ An HTML table is used to display data in rows and columns.

→ It is made up of different tags, each having a specific purpose.

- **<table>**
  - It defines the start and end of the table.
  - All rows, columns, and data are written inside it.
- **<tr> (Table Row)**
  - It represents a single row in the table.
  - Inside a row, we put table headings (<th>) or data cells (<td>).
- **<th> (Table Header Cell)**
  - Used for headings of columns or rows.
  - Text inside <th> is **bold and centered** by default.
- **<td> (Table Data Cell)**
  - Represents normal data in the table.
  - Each <td> is a cell inside a row.
- **<thead> (Table Head Section)**
  - Groups all the header rows of a table.
  - It is useful for styling or when tables are long, so headings stay separate.

Question 2:

What is the difference between colspan and rowspan in tables? Provide examples.

→ **Colspan** and **rowspan** are HTML attributes that "merge" or stretch a single cell across multiple columns or rows in a table.

Difference between colspan and rowspan

1. **colspan**
    a. Used to **merge columns (left to right)**.
    b. A cell will take space of **multiple columns**.
    c. Colspan merges columns **(horizontally).**

➢ **Example:**
<th colspan="2">Name</th>

→ “Name” covers 2 columns.

2. **rowspan**
    a. Used to **merge rows (top to bottom)**.
    b. A cell will take space of **multiple rows**.
    c. Rowpan merges rows **(vertically).**

➢ **Example:**
<th rowspan="2">Subject</th>

→ “Subject” covers 2 rows.

Question 3:

Why should tables be used sparingly for layout purposes? What is a better alternative?

→ Tables were traditionally used to arrange web page layouts, but this is **not recommended** now because:

- They make code **complex and harder to maintain**.
- Tables are meant for **tabular data**, not for page design.
- They reduce **accessibility** for screen readers.
- They are **less flexible** for modern responsive design (mobiles, tablets, etc.).

→ **Better Alternative**

The better alternative is **CSS (Cascading Style Sheets)** with elements like:

- **<div> + CSS** → for structure and design.
- **Flexbox** → for flexible, responsive layouts.
- **CSS Grid** → for advanced, grid-based layouts.

In short: Use **tables only for data**. For layout/design, use **CSS (Flexbox/Grid)**.

We should avoid using HTML <table> tags for page layout because it's the wrong tool for the job. A much better alternative is to use **CSS Flexbox** and **CSS Grid**.