1. Refactor the Code (Debugging)

The following Tailwind CSS code has styling issues on a card component. Identify **two problems** and provide a refactored version.

```
<div class="bg-blue-500 p-4 rounded-lg text-white w-500">
  <h2 class="text-2xl">Welcome</h2>
  <p class="mt-4 text-lg">This is a card.</p>
</div>
```

- **Problem 1**: _____
- **Problem 2**: _____
- **Refactored Code**:

Ans:

<h2 classname=" text-2xl "> Welcome </h2>

 This is a card

2. Multiple Choice (Design Decision)

You're styling a button with Tailwind CSS for a mobile-first app. Which approach best ensures responsiveness?

a) `<button class="px-4 py-2 text-sm md:text-base md:px-6 md:py-3">Click Me</button>`

b) `<button class="p-4 text-lg">Click Me</button>`

c) `<button class="px-6 py-3 text-base">Click Me</button>`

d) `<button class="p-2 md:p-4 text-sm">Click Me</button>`

Answer: ____

Option: A

3. True or False (Coding Standards)

a) Tailwind CSS encourages using utility classes directly in HTML to keep styles predictable and maintainable.

Answer: _____

b) It's a good practice to define custom Tailwind classes in a separate CSS file for reusability.

Answer: _____

3. True (We are using Tailwindcss version 4 so utility class is in built.

4. True

4. Fill in the Blank (Performance)

To reduce Tailwind CSS bundle size in production, you should enable _____ in the `tailwind.config.js` file.

Answer: _____

5.Text white bg- red

6. a-2, b-1, c-3

7. Write the tailwindcss code into tages/ If there is repeated styles create a commmon css file and import (enable the content option in tailwind.config.js)

8

9. 2,4,6 on separete line

10.

a.True

b..True

11. useeffect

12.20

13. a-2 b-1 c-3

14. usestate

15. A(Data is not an aobject)

16. code : const Active= () => {

```
 const box = document.getElementById('box');
 box.classList.toggle('active');
};
```

17. import { useState } from 'react';

```
function Counter() {
 const [count, setCount] = useState(0);
 return (
  <div>
   <p>Count: {count}</p>
   <button onClick={() => setCount(count + 1)}>Increment</button>
  </div>
 );
```

```
}

export default Counter;
```

18.a

19. a. True

b. False

20. Parent

21. World

22 a-2,b-1,c-3

23. React.memo

24. a

25.
```
import { useState, useEffect } from 'react';
 function UsersList() {
  const [users, setUsers] = useState([]);
  const [isLoading, setIsLoading] = useState(true);
  const [error, setError] = useState(null);
   useEffect(() => {
    const fetchUsers = async () => {
      try {
        const response = await fetch('https://jsonplaceholder.typicode.com/users');
        if (!response.ok) throw new Error('Failed to fetch users');
        const data = await response.json();
        setUsers(data);
      } catch (err) {
```

```
        setError(err.message);
      } finally {
        setIsLoading(false);
      }
    };
    fetchUsers();
  }, []);

  if (isLoading) return <div className="text-center p-4">Loading...</div>;
  if (error) return <div className="text-red-500 text-center p-4">Error: {error}</div>;

  return (
    <ul className="max-w-md mx-auto p-4 space-y-2">
      {users.map(user => (
        <li
          key={user.id}
          className="bg-white p-3 rounded-lg shadow hover:bg-gray-50 transition"
        >
          {user.name}
        </li>
      ))}
    </ul>
  );
}
```

```
export default UsersList;
```

```
</div>
```

a) Red background, white text