1. What is Mutable String in Java Explain with an example.

   In Java, String is immutable, meaning once a String object is created, it cannot be modified. However, Java provides two classes, StringBuilder and StringBuffer, which allow mutable (modifiable) strings. These classes are more efficient for operations that require frequent modifications of strings, such as appending, deleting, or updating characters.

   Example :

```
public class Main {
    public static void main(String[] args) {
        // Using StringBuilder for mutable string
        StringBuilder sb = new StringBuilder("Hello");
        System.out.println("Original String: " + sb);

        // Modifying the String
        sb.append(" World");
        System.out.println("Modified String: " + sb);
    }
}
```

2. WAP to reverse a String
   Input : “PWSKILLS”
   Output : “SLLIKSPW”

```
public class Main {
    public static void main(String[] args) {
        String input = "PWSKILLS";
        String reversed = new StringBuilder(input).reverse().toString();
        System.out.println("Reversed String: " + reversed);
    }
}
```

3. WAP to reverse a sentence while preserving the position
   Input : Think Twice
   Output : “kniht eciwt”

```
public class Main {
    public static void main(String[] args) {
        String sentence = "Think Twice";
        String[] words = sentence.split(" ");
        StringBuilder result = new StringBuilder();

        for (String word : words) {
            result.append(new StringBuilder(word).reverse()).append(" ");
        }

        System.out.println("Reversed Sentence: " + result.toString().trim());
    }
}
```

4. WAP to sort a String Alphabetically

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        String input = "PWSKILLS";
        char[] chars = input.toCharArray();
        Arrays.sort(chars);
        String sorted = new String(chars);
        System.out.println("Sorted String: " + sorted);
    }
}
```