1.What is Mutable String in Java Explain with an example.
Ans:
Suppose a String object is created. Now, if any operation is performed on it to alter the String object’s value and those changes are reflected in the same object and no new object is created then such type of a String is called as Mutable String.
Example:

```
public class Test {
    public static void main(String[] args){
        StringBuffer s = new StringBuffer("pw");
        // above line will create a String object in Heap Area and s will be referring to it
        System.out.println(s); // This will print pw
        s.append(" java");
        // Above line will append java to the same String object and the reference will not change
        System.out.println(s); // This will print pw java
    }
}
```

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

```
public class Reverse {
    public static void main(String[] args){
        StringBuffer s = new StringBuffer("PWSKILLS");
        s.delete(0, 2); // delete the first two characters from the string
        s.reverse();    // now reverse the string
        s.append("PW"); // append “PW” to the reversed string
        System.out.println(s);
    }
}
```

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

```
public class ReverseSentencePresevePosition {
    public static void main(String[] args) {
        StringBuilder s = new StringBuilder("Think Twice");
        s.replace(0, 1, "t");
        s.replace(6, 7, "t");
        System.out.println(s);
        StringBuilder s1 = new StringBuilder(s.substring(0, 5));
        StringBuilder s2 = new StringBuilder(s.substring(6, 11));
        s1.reverse();
        s2.reverse();
        StringBuilder sRes = new StringBuilder();
        sRes.append(s1).append(" ").append(s2);
        System.out.println(sRes);
    }
}
```

4. WAP to sort a String Alphabetically.
Ans:

```
public class SortStringAlphabetically {
    // Here, I have taken a string in all lowercases and it does not have
any
    // spaces between the characters
    public static void main(String[] args) {
        String s = new String("pwjavadoesnothaveanyspace");
        char ch[] = s.toCharArray(); // Convert to char array
        int arr[] = new int[26]; // create an int array to capture the
characters
        for(int i=0; i<ch.length; ++i)
            arr[ch[i]-97]++;
        // Create a new String res which will be the result
        StringBuilder res = new StringBuilder();
        for(int i=0; i<arr.length; ++i){
            // Now, depending on how many times a character has occured,
build the resultant String
            int j= arr[i];
            while(j>0){
                char ch1 = (char)(i+97);
                res.append(ch1);
                --j;
            }
        }
        System.out.println(res);
    }
}
```