51.Create a method that finds the first non-repeating character in a String. Input: "swiss" Output: 'w'

```java
public class Main {
    public static char firstNonRepeatingChar(String str) {
        for (int i = 0; i < str.length(); i++) {
            char c = str.charAt(i);
            // Check if the character is unique in the string
            if (str.indexOf(c) == str.lastIndexOf(c)) {
                return c;
            }
        }
        return '\0'; // Return null character if no non-repeating character found
    }

    public static void main(String[] args) {
        String input = "swiss";
        char result = firstNonRepeatingChar(input);

        if (result != '\0') {
            System.out.println("Input: \"" + input + "\"");
            System.out.println("Output: '" + result + "'");
        } else {
            System.out.println("No non-repeating character found.");
        }
    }
}
```

52.Implement a method to compress a String using the counts of repeated characters. If the compressed String is not smaller than the original, return the original String. Input: "aabcccccaaa" Output: "a2b1c5a3"

```
public class StringCompressor {
    public static String compressString(String str) {
        String result = "";
        int count = 1;

        for (int i = 0; i < str.length(); i++) {
            // Count consecutive characters
            if (i + 1 < str.length() && str.charAt(i) == str.charAt(i + 1)) {
                count++;
            } else {
                // Add character and count to result
                result += str.charAt(i) + String.valueOf(count);
                count = 1; // Reset count
            }
        }

        // Return original string if compressed is not smaller
        return result.length() < str.length() ? result : str;
    }

    public static void main(String[] args) {
        String input = "aabcccccaaa";
        String output = compressString(input);
        System.out.println("Input: \"" + input + "\"");
        System.out.println("Output: \"" + output + "\"");
    }
}
```

53.Write a Java program that appends the string " World" to an existing StringBuffer containing "Hello". Input: "Hello"

```java
import java.util.Scanner;

public class AppendToStringBuffer {
    public static void main(String[] args) {
        // Take input from the user
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the initial string: ");
        String input = scanner.nextLine();

        // Initialize StringBuffer with user input
        StringBuffer stringBuffer = new StringBuffer(input);

        // Append " World" to the StringBuffer
        stringBuffer.append(" World");

        // Output the result
        System.out.println("Updated String: " + stringBuffer);

        // Close the scanner
        scanner.close();
    }
}
```

54.Create a method that inserts the string "Beautiful " at index 6 in the StringBuffer containing "Hello World". Input: "Hello World" Output: "Hello Beautiful World"

```java
public class InsertString {
    public static void main(String[] args) {
        // Create a StringBuffer with "Hello World"
        StringBuffer stringBuffer = new StringBuffer("Hello World");

        // Insert "Beautiful " at index 6
        stringBuffer.insert(6, "Beautiful ");

        // Print the updated StringBuffer
        System.out.println(stringBuffer);
    }
}
```

55.Write a Java program that reverses the contents of a StringBuffer initialized with "Java Programming". Input: "Java Programming" Output: "gnimmargorPavaJ"

```
public class ReverseStringBuffer {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "Java Programming"
        StringBuffer stringBuffer = new StringBuffer("Java Programming");

        // Reverse the contents of the StringBuffer
        stringBuffer.reverse();

        // Print the reversed StringBuffer
        System.out.println(stringBuffer);
    }
}
```

56.Create a method that deletes a substring from a StringBuffer.

For example, remove "World" from "Hello World".

Input: "Hello World"

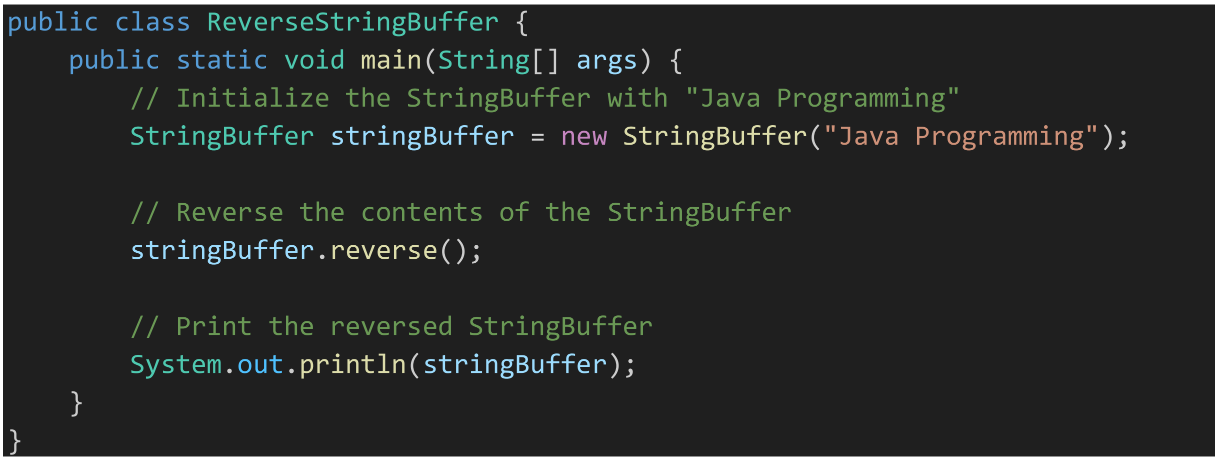
Output: "Hello "

```
public class DeleteSubstring {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "Hello World"
        StringBuffer stringBuffer = new StringBuffer("Hello World");

        // Delete the substring "World" starting from index 6
        stringBuffer.delete(6, 11);

        // Print the updated StringBuffer
        System.out.println(stringBuffer);
    }
}
```

57.Write a program that initializes a StringBuffer with "Java Programming" and reverses its content. Input: "Java Programming" Output: "gnimmargorP avaJ"

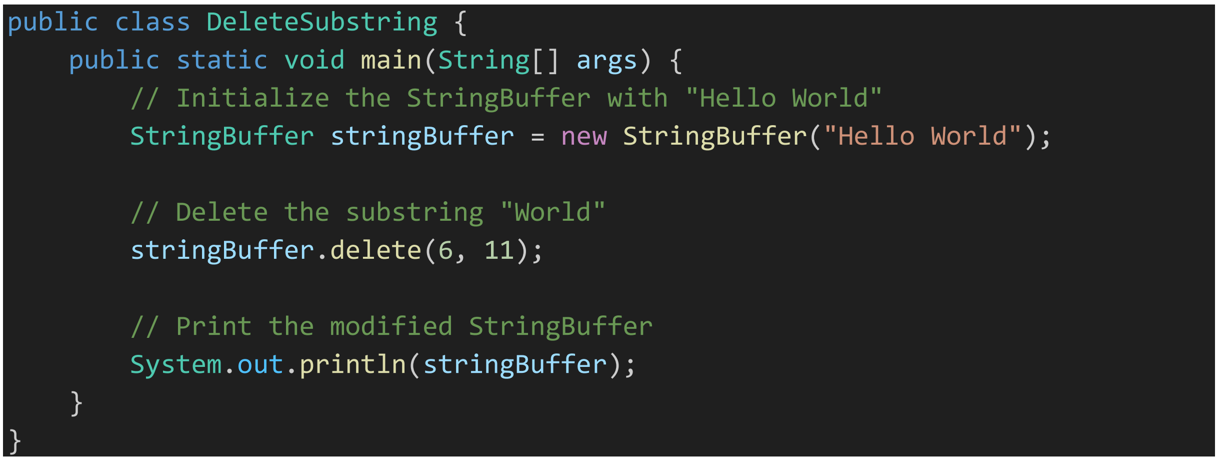


```
public class ReverseStringBuffer {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "Java Programming"
        StringBuffer stringBuffer = new StringBuffer("Java Programming");

        // Reverse the contents of the StringBuffer
        stringBuffer.reverse();

        // Print the reversed StringBuffer
        System.out.println(stringBuffer);
    }
}
```

58.Create a method that deletes the substring "World" from a StringBuffer initialized with "Hello World". Print the modified StringBuffer. Input: "Hello World" Output: "Hello "



```
public class DeleteSubstring {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "Hello World"
        StringBuffer stringBuffer = new StringBuffer("Hello World");

        // Delete the substring "World"
        stringBuffer.delete(6, 11);

        // Print the modified StringBuffer
        System.out.println(stringBuffer);
    }
}
```

59.Write a Java program that replaces "Java" with "Python" in a StringBuffer initialized with "I love Java programming". Input: "I love Java programming" Output: "I love Python programming"

```java
public class ReplaceSubstring {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "I love Java programming"
        StringBuffer stringBuffer = new StringBuffer("I love Java programming");

        // Replace "Java" with "Python"
        int startIndex = stringBuffer.indexOf("Java");
        if (startIndex != -1) {
            stringBuffer.replace(startIndex, startIndex + "Java".length(), "Python");
        }

        // Print the modified StringBuffer
        System.out.println(stringBuffer);
    }
}
```

60.Write a program that creates a StringBuffer, checks its initial capacity, and then appends enough characters to exceed that capacity. Print the new capacity. Input: Initial capacity of StringBuffer Output: New capacity after appending characters

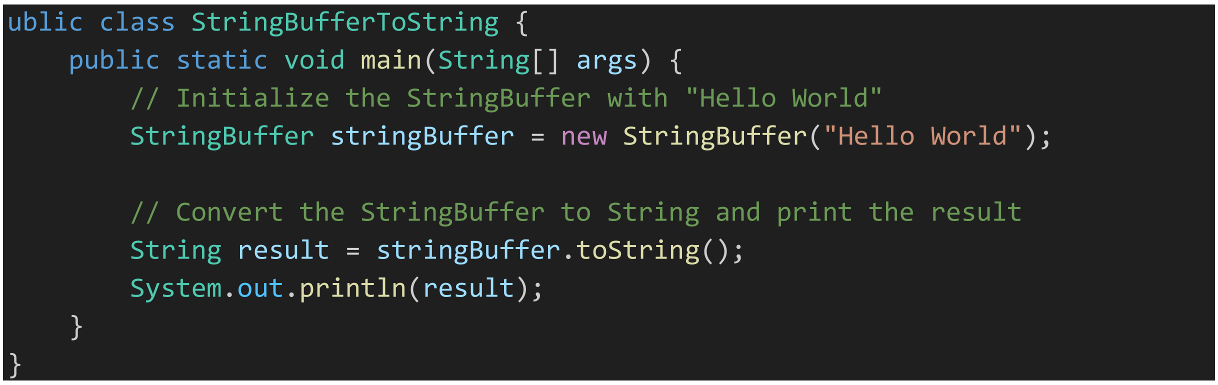
```java
public class StringBufferCapacity {
    public static void main(String[] args) {
        // Create a StringBuffer with an initial capacity of 10
        StringBuffer stringBuffer = new StringBuffer(10);

        // Print the initial capacity
        System.out.println("Initial Capacity: " + stringBuffer.capacity());

        // Append characters to exceed the initial capacity
        stringBuffer.append("HelloWorld123456");

        // Print the new capacity after appending characters
        System.out.println("New Capacity: " + stringBuffer.capacity());
    }
}
```

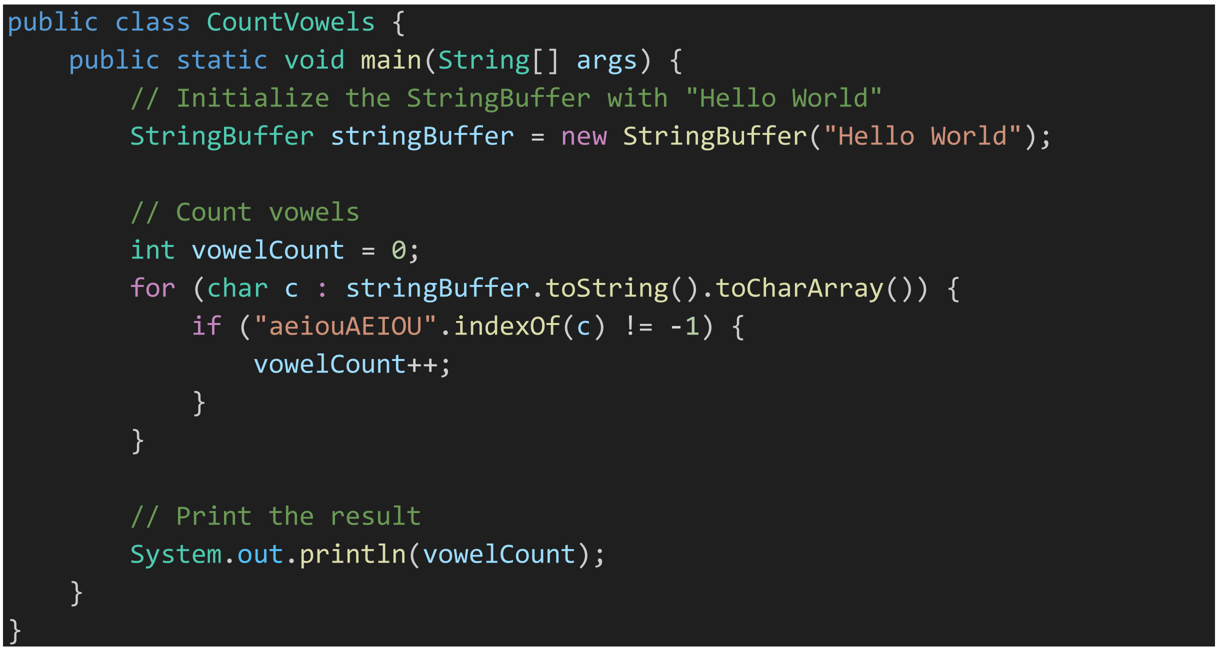
61.Write a method that converts a StringBuffer to a String and returns it. Initialize a StringBuffer with "Hello World" and use your method to print the resulting string. Input: StringBuffer initialized with "Hello World" Output: "Hello World"



```
ublic class StringBufferToString {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "Hello World"
        StringBuffer stringBuffer = new StringBuffer("Hello World");

        // Convert the StringBuffer to String and print the result
        String result = stringBuffer.toString();
        System.out.println(result);
    }
}
```

62.Create a method that counts the number of vowels in a StringBuffer. Initialize it with any string and print the number of vowels. Input: "Hello World" Output: 3



```
public class CountVowels {
    public static void main(String[] args) {
        // Initialize the StringBuffer with "Hello World"
        StringBuffer stringBuffer = new StringBuffer("Hello World");

        // Count vowels
        int vowelCount = 0;
        for (char c : stringBuffer.toString().toCharArray()) {
            if ("aeiouAEIOU".indexOf(c) != -1) {
                vowelCount++;
            }
        }

        // Print the result
        System.out.println(vowelCount);
    }
}
```

63.Write a Java program that initializes a StringBuffer with extra spaces (e.g., " Hello World ") and trims the whitespace from both ends. Input: " Hello World " Output: "Hello World"

```java
public class TrimStringBuffer {
    public static void main(String[] args) {
        // Initialize the StringBuffer with extra spaces
        StringBuffer stringBuffer = new StringBuffer("   Hello World   ");

        // Trim the whitespace from both ends using toString() and trim()
        stringBuffer = new StringBuffer(stringBuffer.toString().trim());

        // Print the result
        System.out.println(stringBuffer);
    }
}
```

64.Create a method that takes two StringBuffer objects and merges them into one, separating them with a space. Print the resulting StringBuffer. Input: StringBuffer1: "Hello", StringBuffer2: "World" Output: "Hello World"

```java
public class MergeStringBuffers {
    public static void main(String[] args) {
        // Initialize two StringBuffer objects
        StringBuffer stringBuffer1 = new StringBuffer("Hello");
        StringBuffer stringBuffer2 = new StringBuffer("World");

        // Merge the two StringBuffers with a space in between
        stringBuffer1.append(" ").append(stringBuffer2);

        // Print the merged StringBuffer
        System.out.println(stringBuffer1);
    }
}
```