String / StringBuffer:

1. Write a Program to check whether a given String is Palindrome or not

PROGRAM:

```
public class Palindrome{
    public static void main(String[] args) {
        String str="madam";
        String rev="";
        for (int i=str.length()-1;i>=0;i--) {
            rev=rev+str.charAt(i);
        }
        if (str.equals(rev)) {
            System.out.println(str + " is a palindrome");
        } else {
            System.out.println(str + " is not a palindrome");
        }
    }
}
```

2. Write a java program that will concatenate 2 strings and return the result. The result should be in lowercase.

Note:If the concatenation creates a double-char, then one of the characters need to be omitted.

Example1)
i/p:Sachin,Tendulkar
o/p:sachin tendulkar
Example2)
i/p:Mark,kate
o/p:markate

PROGRAM:

```
public class ConcatenateTestCases {
    public static String concatenate(String s1, String s2) {
        s1 = s1.toLowerCase();
        s2 = s2.toLowerCase();
        if (!s1.isEmpty() && !s2.isEmpty() &&
            s1.charAt(s1.length() - 1)==s2.charAt(0)) {
            s2 = s2.substring(1);
        }
        return s1+ " " +s2;
    }
    public static void main(String[] args) {
        String result1=concatenate("Sachin","Tendulkar");
         System.out.println("Test Case 1: "+result1);
        String result2=concatenate("Mark","kate");
        System.out.println("Test Case 2: "+result2);
    }
}
```

3.Given a string, return a new string made of 'n' copies of the first 2 chars of the original string where 'n' is the length of the string.

Example1)

i/p:Wipro

o/p:WiWiWiWiWi

PROGRAM:

```
public class Repeat{
    public static String repeat(String str) {
        String part=str.length()<2?str:str.substring(0,2);
        StringBuilder result=new StringBuilder();
        for (int i=0;i<str.length();i++) {
            result.append(part);
        }
        return result.toString();
    }
    public static void main(String[] args) {
        String input="Wipro";
        String output=repeat(input);
        System.out.println(output);
    }
}
```

4. Write a java program that will return the first half of the string, if the length of the string is even. It should return null for odd length string.

Example1)

i/p:TomCat

o/p:Tom

Example2)

i/p:Apron

o/p:null

PROGRAM:

```
public class FirstHalfString {
    public static void main(String[] args) {
        String input = "TomCat";   // change input here
        String result = getFirstHalf(input);
        System.out.println("Output: " + result);
    }

    public static String getFirstHalf(String str) {
        if (str.length() % 2 == 0) {
            return str.substring(0, str.length() / 2);
        } else {
            return null;
        }
    }
}
```

5. Write a java program that accepts a string and returns a new string without the first and last character of the input string.
Example1)
i/p:Suman
o/p:uma

PROGRAM:

```
import java.util.Scanner;
public class RemoveFirstLastChar {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        if (input.length() <= 2) {
            System.out.println("Result: ");
        } else{
            String result = input.substring(1, input.length() - 1);
            System.out.println("Result: " + result);
        }
scanner.close();
    }
}
```

6. Given 2 strings, a and b, return a new string of the form short+long+short, with the shorter string on the outside and the longer string on the inside.
The strings will not be the same length, but they may be empty (length 0).
If input is "hi" and "hello", then output will be "hihellohi".

PROGRAM:

```
public class ShortLongShort {
    public static void main(String[] args) {
        String a = "hi";
        String b = "hello";
        String result;
        if (a.length() < b.length()) {
            result = a + b + a;
        } else {
            result = b + a + b;
        }
        System.out.println("Result: " + result);
    }
}
```

7. Given a string, if the first or last chars are 'x', return the string without those 'x' chars, otherwise return the string unchanged.

If the input is "xHix", then output is "Hi".

If the input is "America", then the output is "America".

PROGRAM:

```
public class RemoveXChars {
    public static void main(String[] args) {
        String input1 = "xHix";
        String input2 = "America";
        System.out.println("Output 1: " + removeX(input1));
        System.out.println("Output 2: " + removeX(input2));
    }
    public static String removeX(String str) {
        int len = str.length();
        if (len > 0 && str.charAt(0) == 'x') {
            str = str.substring(1);
            len = str.length();
        }
        if (len > 0 && str.charAt(str.length() - 1) == 'x') {
            str = str.substring(0, str.length() - 1);
        }
        return str;
    }
}
```

8. Write a Java program that accepts a string (with * in it). The program should return a new string in which the following characters are removed- *, the characters that are to the left and right of *

Example1)

i/p:ab*cd

o/p:ad

PROGRAM:

```
public class RemoveStarAndAdjacent {
    public static void main(String[] args) {
        String input = "ab*cd";
        System.out.println("Output: " + removeStarAndAdjacent(input));
    }
    public static String removeStarAndAdjacent(String str) {
        StringBuilder result = new StringBuilder();
        int len = str.length();
        for (int i = 0; i < len; i++) {
            if (str.charAt(i) == '*'
                || (i > 0 && str.charAt(i - 1) == '*')
                || (i < len - 1 && str.charAt(i + 1) == '*')) {
```

```
                continue;
            }
            result.append(str.charAt(i));
        }
        return result.toString();
    }
}
```

9. Given two strings, a and b, print a new string which is made of the following combination-first character of a, the first character of b, second character of a, second character of b and so on. Any characters left, will go to the end of the result.

Example1)

i/p:Hello,World

o/p:HWeolrllod

PROGRAM:

```
public class MergeStrings {
    public static void main(String[] args) {
        String a = "Hello";
        String b = "World";
        StringBuilder result = new StringBuilder();
        int length = Math.max(a.length(), b.length());
        for (int i = 0; i < length; i++) {
            if (i < a.length()) {
                result.append(a.charAt(i));
            }
            if (i < b.length()) {
                result.append(b.charAt(i));
            }
        }
        System.out.println("Output: " + result.toString());
    }
}
```

10. Given a string and an integer n, print a new string made of n repetitions of the last n characters of the string.

You may assume that n is between 0 and the length of the string, inclusive.

Example1)

i/p:Wipro,3

o/p:propropro

PROGRAM:

```
  public class RepeatLastN {
    public static void main(String[] args) {
        String str = "Wipro";
        int n = 3;
        String Chars = str.substring(str.length() - n);
        StringBuilder result = new StringBuilder();
        for (int i = 0; i < n; i++) {
            result.append(Chars);
        }
        System.out.println("Output: " + result.toString());
    }
}
```

11. Given two strings a and b, return a new string, following the rules given below.
If string b occurs in string a, then the new string should concatenate the characters that appear before and after of String b.
Ignore cases where there is no character before or after the word, and a character may be included twice if it is in between two string b's.
Example1)
i/p:abcXY123XYijk,XY
o/p:c13i
Example2)
i/p:XY123XY,XY
o/p:13

PROGRAM:

```
public class ExtractAroundWord {
    public static void main(String[] args) {
        String a = "abcXY123XYijk";
        String b = "XY";
        StringBuilder result = new StringBuilder();
        int index = 0;
        while ((index = a.indexOf(b, index)) != -1) {
            if (index > 0) {
                result.append(a.charAt(index - 1));
            }
            if (index + b.length() < a.length()) {
                result.append(a.charAt(index + b.length()));
            }
            index += b.length();
        }
        System.out.println("Output: " + result.toString());
    }
```