Strings

1. The web is built with HTML strings like "<i>Yay</i>" which draws Yay as italic text. In this example, the "i" tag makes <i> and </i> which surround the word "Yay". Given tag and word strings, create the HTML string with tags around the word.

Sample Input:

```
i Yay
```

Sample Output:

```
<i>Yay</i>
```

Sample Input:

```
i Hello
```

Sample Output:

```
<i>Hello</i>
```

Sample Input:

```
cite Yay
```

Sample Output:

```
"<cite>Yay</cite>
```

Analysis

1. Take the input from the user.
2. Split the input into two parts.

3. Concatenate the two parts with the tags.
4. Display the output.

`Program: HTMLStrings.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class HTMLStrings {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string: ");
        String userInput = sc.nextLine();
        String[] words = userInput.split(" ");
        System.out.println("<"+words[0]+">"
            +words[1]+
            "</"+words[0]+">"
        );
    }
}
```

Output

```
Enter a string:
i Shabari
<i>Shabari</i>
```

---

2. Given a string, return a new string made of 3 copies of the last 2 chars of the original string. The string length will be at least 2.

Sample Input:

```
Hello
```

Sample Output:

```
lololo
```

Sample Input:

```
Java
```

Sample Output:

```
vavava
```

Sample Input:

```
I
```

Sample Output:

```
String Length should be atleast Two
```

Analysis

1. Take the input from the user.
2. Check if the length of the string is greater than or equal to 2.
3. If yes, then take the last two characters of the string.
4. Repeat the last two characters three times.
5. Display the output.

`Program: LastTwoChars.java`

Code

```java
package com.hands_on;

import java.util.Scanner;

public class LastTwoChars {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String s = sc.nextLine();
        String lastTwoChars = s.substring(s.length()-2);
        System.out.println(
            "Last Two Chars: " + lastTwoChars
            + lastTwoChars + lastTwoChars
        );
    }
}
```

Output

```
Enter the string: Hello
Last Two Chars: lololo
```

---

3. Given a string, return true if it ends in "ly".

Sample Input:

```
oddly
```

Sample Output:

```
true
```

Sample Intput:

```
y
```

Sample Output:

```
false
```

Sample Intput:

```
oddy
```

Sample Output:

```
false
```

Analysis

1. Take the input from the user.
2. Check if the string ends with "ly".
3. If yes, then return true.
4. Else, return false.

`Program: EndsInLy.java`

Code

```java
package com.hands_on;

import java.util.Scanner;

public class EndsInLy {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter any string: ");
        String userInput = sc.nextLine();
        if (userInput.endsWith("ly")) {
            System.out.println("Yes");
        }else{
            System.out.println("No");
        }
    }
}
```

Output

```
Enter any string: oddly
Yes
```

4. Given a string, return true if "bad" appears starting at index 0 or 1 in the string, such as with "badxxx" or "xbadxx" but not "xxbadxx". The string may be any length, including 0.

Analysis

1. Take the input from the user.
2. Check if the string contains "bad" at index 0 or 1.
3. If yes, then return true.
4. Else, return false.

`Program: BadAppears.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class BadAppears {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a word: ");
        String word = sc.nextLine();
        word = word.toLowerCase();
        if (word.indexOf("bad") == 0 || word.indexOf("bad") == 1){
            System.out.println("True");
        }else{
            System.out.println("False");
        }
    }
}
```

Output

```
Enter a word:
badxxx
True
```

---

5. Given two strings, append them together (known as "concatenation") and return the result. However, if the concatenation creates a double-char, then omit one of the chars, so "abc" and "cat" yields "abcat".

SampleInput:

```
abc cat
```

SampleOutput:

```
abcat
```

SampleInput:

```
dog cat
```

SampleOutput:

```
dogcat
```

Analysis

1. Take the input from the user.
2. Concatenate the two strings.
3. Check if the last character of the first string is equal to the first character of the second string.
4. If yes, then omit one of the characters.
5. Display the output.

`Program: ConcatenateStrings.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class ConcatenateStrings {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the first string: ");
        String firstString = sc.nextLine();
        System.out.println("Enter the second string: ");
        String secondString = sc.nextLine();
        if (firstString.charAt(
            firstString.length()-1
            ) == secondString.charAt(0)
        ){
            System.out.println(
                "The Concatenated String: "
                + firstString + secondString.substring(1)
            );
        }else{
            System.out.println(
                "The Concatenated String: "
                + firstString + secondString
            );
        }
    }
}
```

Output

```
Enter the first string:
Shabari K
Enter the second string:
K S
The Concatenated String: Shabari K S
```

6. Write a program to encrypt the text “INDIA” to “KPFKC” and decrypt “KPFKC” to get the original string “INDIA”.

Analysis

1. Take the input from the user.
2. Encrypt the text using the given logic.
3. Decrypt the text using the given logic.
4. Display the output.

`Program: EncryptDecrypt.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class EncryptDecrypt {
    static void encrypt(String str){
        System.out.print("Encrypted String: ");
        String encryptedText = "";
        for(char c : str.toCharArray()) {
            System.out.print((char)(c+2));
            encryptedText += (char)(c+2);
        }
        System.out.println();
        decrypt(encryptedText);
    }

    static void decrypt(String str){
        System.out.print("Decrypted String: ");
        for(char c : str.toCharArray()) {
            System.out.print((char)(c-2));
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter text to encrypt: ");
        String text = sc.nextLine();
        encrypt(text);
    }
}
```

Output

```
Enter text to encrypt: INDIA
Encrypted String: KPFKC
Decrypted String: INDIA
```

---

7. Write a Java program to accept a string from the user. Replace all vowels (`a` , `e` , `I` , `o` , `u`) with `z` . If there are no vowels in the string just print the original word with the message `No vowels present`

Analysis

1. Take the input from the user.
2. Check if the string contains any vowels.
3. If yes, then replace all the vowels with `z` .
4. Display the output.

`Program: ReplaceVowels.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class ReplaceVowels {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = sc.nextLine();
        if(
            s.contains("a")|| s.contains("e")
            || s.contains("i") || s.contains("o")
            || s.contains("u") ||s.contains("A")
            || s.contains("E") || s.contains("I")
            || s.contains("O") || s.contains("U")
        ){
            System.out.print(
                "The replaced string: "
                +s.replaceAll("[aeiouAEIOU]", "z")
            );
        }else{
            System.out.println("No vowels present");
        }
    }
}
```

Output

```
Enter a string:  shabari
The replaced string: shzbzrz
```

---

8. Write a java program that takes as input a string as a Sentence and return its second word in uppercase. Note : if the input string(sentence) with less than 2 words, the code should print the word ”LESS”

Sample Input:

```
MontBleu Technologies Bangalore
```

Sample Output:

```
TECHNOLOGIES
```

Sample Input:

```
World Cup
```

Sample Output:

```
CUP
```

Sample Input:

```
Championship 2017 League
```

Sample Output:

```
2017
```

Sample Input:

```
Hello
```

Sample Output

```
LESS
```

Analysis

1. Take the input from the user.
2. Split the input into words.
3. Check if the length of the words is greater than or equal to 2.
4. If yes, then convert the second word to uppercase.
5. Else, print "LESS".
6. Display the output.

`Program: SecondWord.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class SecondWord {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String sentence = sc.nextLine();
        String[] words = sentence.split(" ");
        if (words.length >= 2){
            System.out.println(words[1].toUpperCase());
        }else{
            System.out.println("LESS");
        }
    }
}
```

Output

```
Enter a sentence: I am Shabari
AM
```

---

9. Given two strings a and b. Write a Java function isAnagram () to check whether the two strings are anagrams or not.

> Note: An anagram of a string is another string that contains the same characters, only the order of characters can be different. For example: Listen and Silent are anagrams

Character Frequency:	**Listen Frequency:**	**Silent**
L or l	1	1
I or i	1	1
S or s	1	1
T or t	1	1
E or e	1	1
N or n	1	1

Analysis

1. Take the input from the user.
2. Check if the length of the two strings is equal.
3. If yes, then convert the strings to lowercase.
4. Sort the strings.
5. Check if the two strings are equal.
6. If yes, then return true.
7. Else, return false.

`Program: Anagram.java`

Code

```
package com.hands_on;

import java.util.Arrays;
import java.util.Scanner;

public class Anagram {
    static boolean isAnagram(String a, String b) {
        if (a.length() != b.length()){
            return false;
        }
        a = a.toLowerCase();
        b = b.toLowerCase();
        char[] aArray = a.toCharArray();
        char[] bArray = b.toCharArray();
        Arrays.sort(aArray);
        Arrays.sort(bArray);
        return Arrays.equals(aArray, bArray);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the first string: ");
        String a = sc.nextLine();
        System.out.println("Enter the second string: ");
        String b = sc.nextLine();
        if (isAnagram(a, b)){
            System.out.println("Anagram");
        }else{
            System.out.println("Not Anagram");
        }
    }
}
```

Output

```
Enter the first string: Shabari
Enter the second string: barisha
Anagram
```

---

10. Write a Java Code to search a text and count the occurrence of the search text in the given sentence.

Analysis

1. Take the input from the user.
2. Split the input into words.
3. Count the occurrence of the search text.
4. Display the output.

`Program: CountOccurrence.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class CountOccurrence {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String sentence = sc.nextLine();
        System.out.print("Enter the search text: ");
        String searchText = sc.nextLine();
        String[] words = sentence.split(" ");
        int count = 0;
        for (String word : words){
            if (word.equals(searchText)){
                count++;
            }
        }
        System.out.println("The occurrence of the search text: " + count);
    }
}
```

Output

```
Enter a sentence : I am Shabari is
Enter the search text: is
The occurrence of the search text: 1
```

---

11. Write a Java function that takes a sentence and a string as input and checks if the string is present. If it is, convert lowercase letters to uppercase and uppercase letters to lowercase, and return the converted string

Analysis

1. Take the input from the user.
2. Split the input into words.
3. Check if the string is present in the sentence.
4. If yes, then convert the lowercase letters to uppercase and uppercase letters to lowercase.
5. Display the output.

`Program: ConvertCase.java`

Code

```
package com.hands_on;

import java.util.Scanner;

public class ConvertCase {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String s = sc.nextLine();
        String[] words = s.split(" ");
        System.out.print("Enter the word: ");
        String word = sc.next();
        String convertedString = "";
        for (String w : words){
            if (w.equalsIgnoreCase(word)){
                for (char c : w.toCharArray()){
                    if (Character.isLowerCase(c)){
                        convertedString += Character.toUpperCase(c);
                    }else{
                        convertedString += Character.toLowerCase(c);
                    }
                }
            }else{
                convertedString += w;
            }
            convertedString += " ";
        }
        System.out.println(convertedString.trim());
    }
}
```

Output

```
Enter the string: i am shabari
Enter the word: am
i AM shabari
```

12. Given a string, the task is to reverse the order of the words in the given string.

Analysis

1. Take the input from the user.

2. Split the input into words.
3. Reverse the order of the words.

`Program: ReverseWords.java`

Code

```java
package com.hands_on;

import java.util.Scanner;

public class ReverseWords {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the String: ");
        String s = sc.nextLine();
        String[] words = s.split(" ");
        String reverseString = "";
        for (int i = words.length-1; i >= 0; i--){
            reverseString += words[i] + " ";
        }
        System.out.println("The reverse string is `"+reverseString.trim()+"`"
    }
}
```

Output

```
Enter the String: i am Shabari
The reverse string is `Shabari am i`
```

---

13. Given a list of words followed by two words, the task is to find the minimum distance between the given two words in the list of words.

SampleInput:

```
S = {“the”, “quick”, “brown”, “fox”, “quick”},
word1 = “the”, word2 = “fox”
```

SampleOutput:

```
Minimum distance between the words “the” and “fox” is 3
```

Analysis

1. Take the input from the user.
2. Find the index of the two words.
3. Calculate the minimum distance between the two words.

`Program: MinimumDistance.java`

Code

```java
package com.hands_on;

import java.util.Scanner;

public class MinimumDistance {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of words: ");
        int n = sc.nextInt();
        String[] words = new String[n];
        for (int i = 0; i < n; i++){
            words[i] = sc.next();
        }
        System.out.print("Enter the first word: ");
        String word1 = sc.next();
        System.out.print("Enter the second word: ");
        String word2 = sc.next();
        int index1 = -1, index2 = -1;
        int minDistance = Integer.MAX_VALUE;
        for (int i = 0; i < n; i++){
            if (words[i].equals(word1)){
                index1 = i;
            }
            if (words[i].equals(word2)){
                index2 = i;
            }
            if (index1 != -1 && index2 != -1){
                minDistance = Math.min(minDistance, Math.abs(index1 - index2)
            }
        }
        System.out.println("Minimum distance between the words "+word1+" and
    }
}
```

Output

```
Enter the number of words: 5
the quick brown fox quick
Enter the first word: the
Enter the second word: fox
Minimum distance between the words the and fox is 3
```