B.Hanumanthu-ISTE60

1. Java Program to Convert char to String and String to Char

```
public class StringToChar{

   public static void main(String[] args) {

      // Test the conversion from char to String
      char testChar = 'A';
      String convertedString = charToString(testChar);
      System.out.println("The converted String from char 'A' is: " + convertedString);

      // Test the conversion from String to char
      String testString = "B";
      char convertedChar = stringToChar(testString);
      System.out.println("The converted char from String 'B' is: " + convertedChar);
   }

   // Function to convert char to String
   public static String charToString(char character) {
      return Character.toString(character);
   }

   // Function to convert String to char
   public static char stringToChar(String string) {
      if (string.length() != 1) {
         throw new IllegalArgumentException("The string should contain exactly one
character");
      }
```

```
        return string.charAt(0);
    }
}
```

```
D:\Java>notepad

D:\Java>javac StringToChar.java

D:\Java>java StringToChar
The converted String from char 'A' is: A
The converted char from String 'B' is: B

D:\Java>
```

2. Java Program to find duplicate characters in a String

```
import java.util.HashSet;

public class DuplicateCharacter {
    public static void main(String[] args) {
        String input = "Test Case";
        findDuplicateCharacters(input);
    }

    public static void findDuplicateCharacters(String input) {
        HashSet<Character> characterSet = new HashSet<>();
        for (char c : input.toCharArray()) {
            if (!characterSet.add(c)) {
                System.out.println("Duplicate character: " + c);
            }
        }
    }
}
```

```
D:\Java>javac DuplicateCharacter.java

D:\Java>java DuplicateCharacter
Duplicate character: s
Duplicate character: e

D:\Java>
```

3. Java Program to check Palindrome String using Stack, Queue, For and While loop

Using Stack

```
 import java.util.Stack;


public class Palindrome {


   public static void main(String[] args) {
      String str = "Able was I ere I saw Elba";
      System.out.println(isPalindrome(str));
   }


   public static boolean isPalindrome(String str) {
      Stack<Character> stack = new Stack<>();


      for (char c : str.toCharArray()) {
         if (c != ' ') {
            stack.push(Character.toLowerCase(c));
         }
      }
```

```
        for (char c : str.toCharArray()) {
            if (c != ' ') {
                if (stack.pop() != Character.toLowerCase(c)) {
                    return false;
                }
            }
        }

        return true;
    }
}
```

```
D:\Java>javac Palindrome.java

D:\Java>java Palindrome
true

D:\Java>
```

4. Java Program to sort strings in alphabetical order

```
import java.util.Arrays;
import java.util.Comparator;

public class SortStrings {

    public static void main(String[] args) {

        String[] strArray = {"Ball", "Apple", "Cat", "Dog", "Elephant"};

        System.out.println("Array Before Sorting:");
        printArray(strArray);
```

```
        sortStrings(strArray);

        System.out.println("\nArray After Sorting in Ascending Order:");
        printArray(strArray);
    }

    public static void sortStrings(String[] strArray) {
        Arrays.sort(strArray, new Comparator<String>() {
            @Override
            public int compare(String s1, String s2) {
                return s1.compareToIgnoreCase(s2);
            }
        });
    }

    public static void printArray(String[] strArray) {
        for (String str : strArray) {
            System.out.println(str);
        }
    }
}
```

```
D:\Java>javac SortStrings.java

D:\Java>java SortStrings
Array Before Sorting:
Ball
Apple
Cat
Dog
Elephant

Array After Sorting in Ascending Order:
Apple
Ball
Cat
Dog
Elephant

D:\Java>
```

5. Java Program to reverse words in a String

```
public class ReverseWordsInString {

   public static void main(String[] args) {
      String input = "I am learning Java";
      String reversed = reverseWords(input);
      System.out.println("Reversed words string: " + reversed);
   }

   public static String reverseWords(String s) {
      if (s == null || s.isEmpty()) {
         return s;
      }

      String[] words = s.split(" ");
      StringBuilder reversedString = new StringBuilder();

      for (int i = words.length - 1; i >= 0; i--) {
         reversedString.append(words[i]).append(" ");
      }

      return reversedString.toString().trim();
   }
}
```

```
D:\Java>javac ReverseWordsInString.java

D:\Java>java ReverseWordsInString
Reversed words string: Java learning am I

D:\Java>
```

6. Java Program to perform bubble sort on Strings

```
import java.util.Arrays;

public class StringBubbleSort{
```

```
public static void main(String[] args) {

   String[] strArray = {"Dog", "Cat", "Bird", "Fish", "Lion"};

   System.out.println("Original Array:");
   System.out.println(Arrays.toString(strArray));

   bubbleSort(strArray);

   System.out.println("\nSorted Array:");
   System.out.println(Arrays.toString(strArray));
}

public static void bubbleSort(String[] strArray) {

   int n = strArray.length;
   boolean swapped;

   for (int i = 0; i < n - 1; i++) {

      swapped = false;

      for (int j = 0; j < n - 1 - i; j++) {

         if (strArray[j].compareTo(strArray[j + 1]) > 0) {

            // Swap strArray[j] and strArray[j + 1]
            String temp = strArray[j];
            strArray[j] = strArray[j + 1];
            strArray[j + 1] = temp;

            swapped = true;
         }
      }

      // If the inner loop did not swap any elements, the array is already sorted
      if (!swapped) {
         break;
```

```
        }
      }
    }
}
```

```
D:\Java>javac StringBubbleSort.java

D:\Java>java StringBubbleSort
Original Array:
[Dog, Cat, Bird, Fish, Lion]

Sorted Array:
[Bird, Cat, Dog, Fish, Lion]

D:\Java>
```

7. Java program to find occurrence of a character in a String

```
public class CountOccurences{

   public static void main(String[] args) {

      String str = "Hello World!";
      char charToFind = 'o';

      int count = countOccurrences(str, charToFind);

      System.out.println("The character '" + charToFind + "' occurs " + count + " times in the string.");
   }

   public static int countOccurrences(String str, char charToFind) {

      int count = 0;

      for (int i = 0; i < str.length(); i++) {
         if (str.charAt(i) == charToFind) {
            count++;
         }
      }
```

```
        return count;
    }
}
```

```
D:\Java>javac CountOccurences.java

D:\Java>java CountOccurences
The character 'o' occurs 2 times in the string.

D:\Java>
```

8. Java program to count vowels and consonants in a String

```
public class CountVowelsConsonants{

    public static void main(String[] args) {

        String str = "This is a test String";
        str = str.toLowerCase();

            int con = 0;
            int vow = 0;

            for(int i=0;i<str.length();i++){
                    char ch = str.charAt(i);

                    if(ch=='a'|| ch=='e' || ch=='i' || ch=='o' || ch=='u'){
                            vow++;
                    }
                    else if(ch>='a' && ch<='z'){
                            con++;
                    }
            }
            System.out.println("Vowels in String: "+vow);
            System.out.println("Consonants in String: "+con);
    }
}
```

```
D:\Java>javac CountVowelsConsonants.java

D:\Java>java CountVowelsConsonants
Vowels in String: 5
Consonants in String: 12

D:\Java>
```

9. Java Program to check two strings are anagram or not

```
import java.util.Arrays;

public class CheckAnagram {

   public static void main(String[] args) {

      String str1 = "listen";
      String str2 = "silent";

      System.out.println("Are \"" + str1 + "\" and \"" + str2 + "\" anagrams? " +
areAnagram(str1, str2));
   }

   public static boolean areAnagram(String str1, String str2) {

      // If both strings lengths are not same, return false
      if (str1.length() != str2.length()) {
         return false;
      }

      // Convert the strings to character arrays
      char[] charArray1 = str1.toCharArray();
      char[] charArray2 = str2.toCharArray();

      // Sort the character arrays
      Arrays.sort(charArray1);
      Arrays.sort(charArray2);
```

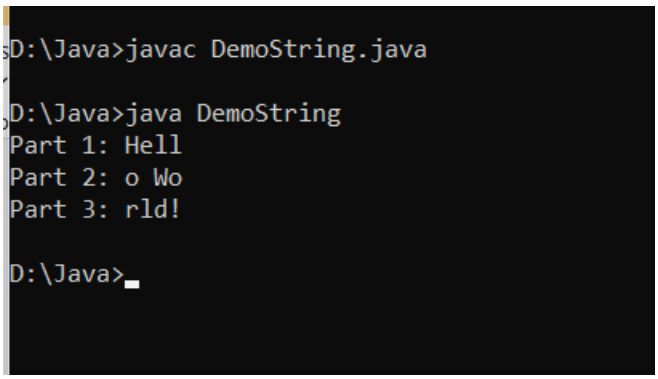
```
        // Compare the sorted character arrays
        return Arrays.equals(charArray1, charArray2);
    }
}
```

```
D:\Java>javac CheckAnagram.java

D:\Java>java CheckAnagram
Are "listen" and "silent" anagrams? true

D:\Java>
```

10. Java Program to divide a string in ‘n’ equal parts

```
public class  DemoString{

    public static void main(String[] args) {

        String str = "Hello World!";
        int n = 3;

        // Calculate the size of each part
        int sizeOfEachPart = str.length() / n;

        // Array to store the parts
        String[] parts = new String[n];

        // Indexes for start and end
        int startIndex = 0;
        int endIndex = 0;

        // Divide the string into 'n' parts
        for (int i = 0; i < n; i++) {
            startIndex = endIndex;
            endIndex += sizeOfEachPart;
            if (i == n - 1) { // if last part
```

```
                endIndex = str.length();
            }
            parts[i] = str.substring(startIndex, endIndex);
        }

        // Print the divided parts
        for (int i = 0; i < parts.length; i++) {
            System.out.println("Part " + (i + 1) + ": " + parts[i]);
        }
    }
}
```



11. Java Program to find all subsets of a string

```
import java.util.Arrays;

public class SubSet{

    public static void main(String[] args) {
        String str = "abc";
        System.out.println("All subsets of the string: " + str);
        getAllSubsets(str);
    }

    private static void getAllSubsets(String str) {
        int n = str.length();
        for (int i = 0; i < (1 << n); i++) {
```

```
            StringBuilder subset = new StringBuilder();
            for (int j = 0; j < n; j++) {
                if ((i & (1 << j)) > 0) {
                    subset.append(str.charAt(j));
                }
            }
            System.out.println(subset.toString());
        }
    }
}
```