Strings in Java
Assignment Questions

Assignment Questions

1. Write a program to remove Duplicates from a String.

Ans →
```
public class RemoveDuplicate{
     public static void main(String[] args) {
        String s = "Hello brother";
        String s2 = "";
        for (int i = 0; i < s.length(); i++) {
           Boolean found = false;
           for (int j = 0; j < s2.length(); j++) {
              if (s.charAt(i) == s2.charAt(j)) {
                 found = true;
                 break; //don't need to iterate further
              }
           }
           if (found == false) {
              s2 = s2.concat(String.valueOf(s.charAt(i)));
           }
        }
        System.out.println(s2);
     }
  }
```

2. Write a program to print Duplicate characters from the string.

Ans →
```
import java.util.*;
  public class demo {
     public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String: ");
        String str = sc.nextLine();
        for(int i=0; i<str.length(); i++) {
           for(int j=i+1; j<str.length(); j++) {
              if(str.charAt(i) == str.charAt(j)) {
                 System.out.println("Duplicate character: " + str.charAt(i));
              }
           }
        }
     }
  }
```

3. Write a program to check if “2552” is a palindrome or not.

Ans →

```
import java.lang.String;
public class Palindrome {
    public static void main(String[] args) {
        String str1 = "2552";
        String str2 = "";
        for(int i = str1.length()-1; i>=0; i--) {
            str2 = str2+str1.charAt(i);
        }
        if(str1.equals(str2)) {
            System.out.println("Given String is Palindrome");
        }
        else {
            System.out.println("Given String is Not Palindrome");
        }
    }
}
```

4. WAP to count the number of consonants, vowels, and special characters in a String.

Ans →

```
import java.util.Scanner;
public class CountConsonantsVowelsSpecialCharacters {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String str = input.nextLine();

        int vowels = 0, consonants = 0, specialChars = 0;

        str = str.toLowerCase();

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

            if (c >= 'a' && c <= 'z') {
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
                    vowels++;
                } else {
                    consonants++;
                }
            } else {
                specialChars++;
            }
        }

        System.out.println("Number of vowels: " + vowels);
        System.out.println("Number of consonants: " + consonants);
        System.out.println("Number of special characters: " + specialChars);
    }
}
```

5. WAP to implement Anagram Checking least inbuilt method being used.

Ans →

```
import java.util.Scanner;
public class demo {
    public static boolean areAnagrams(String str1, String str2) {
        if (str1.length() != str2.length()) {
            return false;
        }
        int[] count = new int[256];

        for (int i = 0; i < str1.length(); i++) {
            count[str1.charAt(i)]++;
            count[str2.charAt(i)]--;
        }
        for (int i = 0; i < 256; i++) {
            if (count[i] != 0) {
                return false;
            }
        }

        return true;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter the first string: ");
        String str1 = input.nextLine();

        System.out.print("Enter the second string: ");
        String str2 = input.nextLine();

        if (areAnagrams(str1, str2)) {
            System.out.println("The two strings are anagrams.");
        } else {
            System.out.println("The two strings are not anagrams.");
        }
    }

}
```

6. WAP to implement Pangram Checking least inbuilt method being used.

Ans →

```
import java.util.Scanner;
public class PangramChecking {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String str = input.nextLine();

        if (isPangram(str)) {
            System.out.println("The string is a pangram.");
        } else {
            System.out.println("The string is not a pangram.");
        }
    }

    public static boolean isPangram(String str) {
        int[] count = new int[26];

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

            if (c >= 'A' && c <= 'Z') {
                count[c - 'A']++;
            } else if (c >= 'a' && c <= 'z') {
                count[c - 'a']++;
            }
        }

        for (int i = 0; i < 26; i++) {
            if (count[i] == 0) {
                return false;
            }
        }

        return true;
    }
}
```

7. WAP to find if the String contains all unique characters.

Ans →

```
import java.util.Scanner;
public class UniqueCharacters {

    public static boolean hasAllUniqueChars(String str) {
        boolean[] charSet = new boolean[256];

        for (int i = 0; i < str.length(); i++) {
            int val = str.charAt(i);
            if (charSet[val]) {
                return false;
            }

            charSet[val] = true;
        }

        return true;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String str = input.nextLine();

        if (hasAllUniqueChars(str)) {
            System.out.println("The string contains all unique characters.");
        } else {
            System.out.println("The string does not contain all unique characters.");
        }
    }

}
```

8. WAP to find the maximum occurring character in a String.

Ans →

```
import java.util.Scanner;
public class demo {
    public static char getMaxOccurringChar(String str) {
        int[] count = new int[256];

        for (int i = 0; i < str.length(); i++) {
            int val = str.charAt(i);
            count[val]++;
        }

        int maxCount = 0;
        char maxChar = ' ';

        for (int i = 0; i < 256; i++) {
            if (count[i] > maxCount) {
                maxCount = count[i];
                maxChar = (char) i;
            }
        }

        return maxChar;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String str = input.nextLine();

        char maxChar = getMaxOccurringChar(str);

        System.out.println("The maximum occurring character is: " + maxChar);
    }


}
```