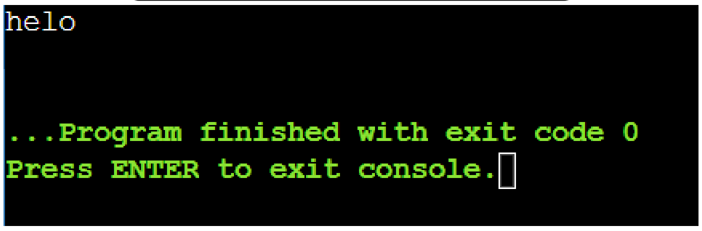
Assignment – 13th (Strings in Java)

1. Write a program to remove Duplicates from a String. (Take any String example with duplicates character)

Ans: - Here is a simple program in Java that remove duplicates from a string:

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

        String str1 = "hello";
        StringBuilder str2 = new StringBuilder();

        for(int i=0; i<str1.length(); i++) {

            String str3 = Character.toString(str1.charAt(i));

            if(str2.indexOf(str3) == -1) {
                str2.append(str3);
            }
        }

        System.out.println(str2.toString());
    }
}
```

OUTPUT



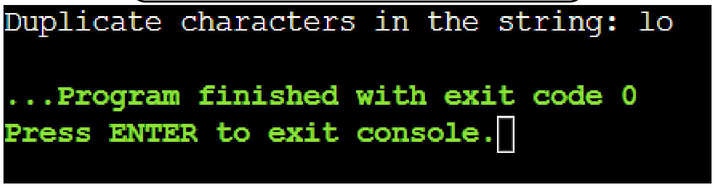
```
helo


...Program finished with exit code 0
Press ENTER to exit console.
```

2. WAP to print Duplicates characters from the String.

Ans: - Here is a simple program in Java that print duplicates from the string:

```
public class Main {
    public static void main(String[] args) {
        String str = "Hello World";
        int count;

        char[] stringArray = str.toCharArray();

        System.out.print("Duplicate char in the string: ");

        for(int i=0; i<stringArray.length; i++) {
            count = 1;

            for(int j=i+1; j<stringArray.length; j++) {
                if(stringArray[i] == stringArray[j] &&
                   stringArray[i] != ' ') {
                   count++;
                   stringArray[j] = '0';
                }
            }
            if(count > 1 && stringArray[i] != '0'){
                System.out.print(stringArray[i]);
             }
        }
    }
}
```

OUTPUT



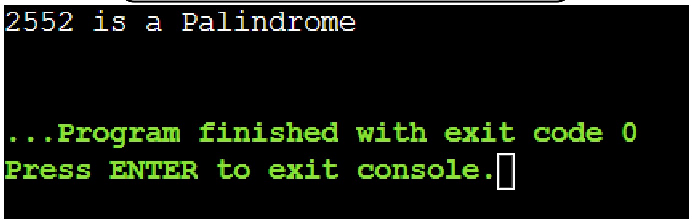
```
Duplicate characters in the string: lo

...Program finished with exit code 0
Press ENTER to exit console.
```

3. WAP to check if “2552” is palindrome or not.

Ans: - Check number is palindrome or not:

```
public class Main
{
    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(str1 + " is a Palindrome");
        }
        else{
            System.out.println(str1 + " is not a Palindrome");
        }
    }
}
```

OUTPUT



```
2552 is a Palindrome


...Program finished with exit code 0
Press ENTER to exit console.
```

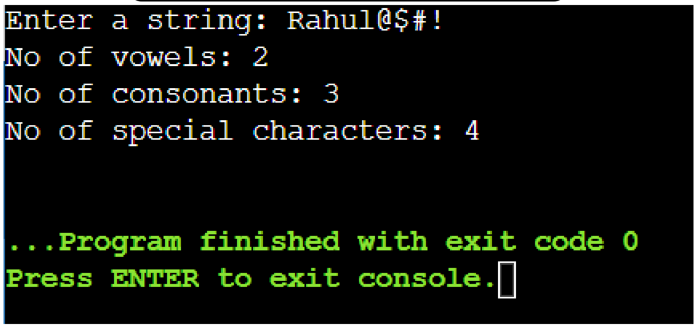
4. WAP to count the number of consonants, vowels, special characters in a String.

Ans: - Count the number of constants, vowels, special characters in a String:

```
import java.util.Scanner;
```

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

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine();

        int vowels=0, consonants=0, specialChars=0;
        str = str.toLowerCase();

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

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

        System.out.println("No of vowels: " + vowels);
        System.out.println("No of consonants: " + consonants);
        System.out.println("No of special characters: " +
specialChars);
    }
}
```

OUTPUT



```
Enter a string: Rahul@$#!
No of vowels: 2
No of consonants: 3
No of special characters: 4


...Program finished with exit code 0
Press ENTER to exit console.
```

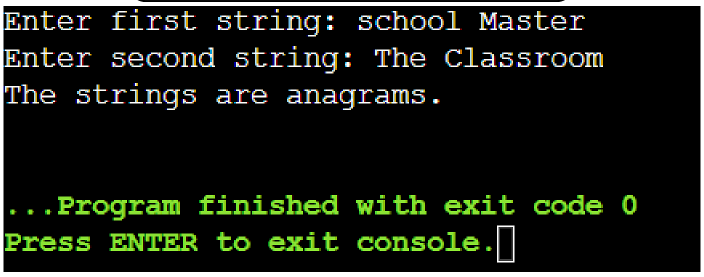
5. WAP to implement Anagram Checking with least inbuilt methods being used.

Ans: - Here is a program in Java to implement Anagram:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String s1 = input.nextLine();
```

```java
        System.out.print("Enter second string: ");
        String s2 = input.nextLine();

        boolean areAnagrams = true;

        // Check if strings are of equal length
        if(s1.length() != s2.length()) {
            areAnagrams = false;
        }else {
            // Convert strings to character arrays
            char[] s1Array = s1.toLowerCase().toCharArray();
            char[] s2Array = s2.toLowerCase().toCharArray();

            // Sort character arrays
            for(int i=0; i<s1Array.length; i++) {
                for(int j= i+1; j<s1Array.length; j++) {
                    if(s1Array[i] > s1Array[j]) {
                        char temp = s1Array[i];
                        s1Array[i] = s1Array[j];
                        s1Array[j] = temp;
                    }
                    if(s2Array[i] > s2Array[j]) {
                        char temp = s2Array[i];
                        s2Array[i] = s2Array[j];
                        s2Array[j] = temp;
                    }
                }
            }

            // Compare character arrays
            for(int i=0; i<s1Array.length; i++) {
                if(s1Array[i] != s2Array[i]) {
                    areAnagrams = false;
                    break;
                }
            }
        }

        if(areAnagrams) {
            System.out.println("The strings are anagrams.");
        }else {
            System.out.println("The strings are not anagrams.");
        }
    }
}
```

OUTPUT



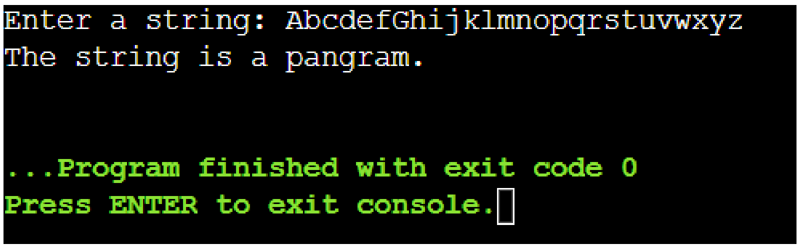
```
Enter first string: school Master
Enter second string: The Classroom
The strings are anagrams.


...Program finished with exit code 0
Press ENTER to exit console.
```

6. WAP to implement Pangram Checking with least inbuilt methods being used.

Ans: - Here's a Java program to implement Pangram:

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = input.nextLine();

        boolean isPangram = true;

        // Convert string to lowercase
        s = s.toLowerCase();

        // Create a boolean array to store character presence
        boolean[] present = new boolean[26];

        // Iterate over each character of the string
        for(int i=0; i<s.length(); i++) {
            char c = s.charAt(i);

            // Check if character is an English alphabet
            if(c >= 'a' && c <= 'z') {
                // Set presence of character in the array
                present[c - 'a'] = true;
            }
        }

        // Check if all alphabets are present in the string
        for(int i=0; i<present.length; i++) {
            if(!present[i]) {
                isPangram = false;
                break;
            }
        }

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

OUTPUT



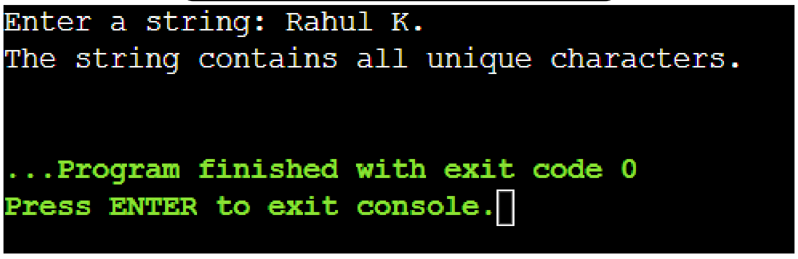
```
Enter a string: AbcdefGhijklmnopqrstuvwxyz
The string is a pangram.


...Program finished with exit code 0
Press ENTER to exit console.
```

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

Ans: - Here's a Java program to check if string contains all unique characters:

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

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

        boolean uniqueChars = true;

        // Create a boolean array to store character presence
        boolean[] present = new boolean[256];

        // Iterate over each character of the string
        for(int i=0; i<s.length(); i++) {
            char c = s.charAt(i);

           // Check if character is already present in the array
            if(present[c]) {
                uniqueChars = false;
                break;
            }else {
                // Set presence of character in the array
                present[c] = true;
            }
        }

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

OUTPUT



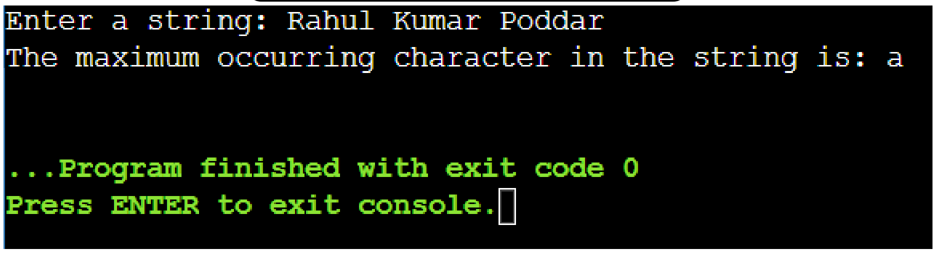
```
Enter a string: Rahul K.
The string contains all unique characters.


...Program finished with exit code 0
Press ENTER to exit console.
```

8. WAP to find the maximum occurring characters in a string.

Ans: - Here's a Java program to check if string contains all unique characters:

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

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

        // Create an array to store character frequency
        int[] freq = new int[256];

        // Iterate over each character of the string and update
frequency
        for(int i=0; i<s.length(); i++) {
            char c = s.charAt(i);
            freq[c]++;
        }

        // Find the character with maximum frequency
        char maxChar = 0;
        int maxFreq = 0;

        for(int i=0; i<freq.length; i++) {
            if(freq[i] > maxFreq) {
                maxChar = (char) i;
                maxFreq = freq[i];
            }
        }

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

OUTPUT



```
Enter a string: Rahul Kumar Poddar
The maximum occurring character in the string is: a


...Program finished with exit code 0
Press ENTER to exit console.
```