ASSIGNMENT
STRINGS

Q.1 WAP to remove duplicates from a string

```java
package StringsAssignment;
public class ques1 {

    public static void main(String[] args) {
        String str = "coding in java";
        String ans="";

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

        System.out.println(ans);
    }
}
```

Console ×

<terminated> ques1 (5) [Java Application] C:\Users\dell6\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\javaw.

```
coding jav
```

Q.2 WAP to duplicate characters from the string.

```java
package StringsAssignment;
public class ques2 {
    public static void main(String[] args) {
        String str = "Discrimination";
        String uniqueString = "";
        String duplicateString ="";
        for(int i=0;i<str.length();i++) {
            char c = str.charAt(i);
            if(uniqueString.indexOf(c)<0) {
                uniqueString+=c;
            }else {
                if(duplicateString.indexOf(c)<0) {
                    duplicateString+=c;
                }
            }
        }
        for(int i=0;i<duplicateString.length();i++) {
            System.out.print(duplicateString.charAt(i)+" ");
        }
    }
}
```

Console ×

<terminated> ques2 (5) [Java Application] C:\Users\dell6\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\javaw.exe (15-Feb-2023, 11:35:25 am – 11:35:2

```
i n
```

Q.3 WAP to check if "2552" is Palindrome or not.

```java
package StringsAssignment;
public class ques3 {
    public static void main(String[] args) {
        String num = "2552";
        int i = 0;
        int j = num.length()-1;

        while(i<j) {
            if(num.charAt(i)!=num.charAt(j)) {
                System.out.println("Not a pallindrome...");
                break;
            }
            i++;
            j--;
        }
        System.out.println("It's a pallindrome");
    }
}
```

Console ×
<terminated> ques3 (6) [Java Application] C:\Users\dell6\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\javaw.exe (15-Feb-2023, 11:38:21 am – 11
```
It's a pallindrome
```

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

```java
package StringsAssignment;
public class ques4 {
    public static void main(String[] args) {
        // consonants , vowels , special characters
        String str = "This is a java * program *";
        int consonants = 0;
        int vowels = 0 ;
        int specialCharacters = 0;

        for(int i=0;i<str.length();i++) {
            if((str.charAt(i)>='a' && str.charAt(i)<='z')||
                (str.charAt(i)>='A' && str.charAt(i)<='Z')) {
                // vowels
                if(str.charAt(i)=='a' || str.charAt(i)=='e'|| str.charAt(i)=='i'
                    || str.charAt(i)=='o' || str.charAt(i)=='u' ||
                    str.charAt(i)=='A' || str.charAt(i)=='E' || str.charAt(i)=='I'
                    || str.charAt(i)=='O' || str.charAt(i)=='U') {
                    vowels++;
                }else {
                    consonants++;
                }
```

```java
21              }
22              }else if(str.charAt(i)==' '){   // space
23                  continue;
24              }else {    // special characters
25                  specialCharacters++;
26              }
27          }
28          System.out.println("Consonants: "+consonants);
29          System.out.println("Vowels: "+vowels);
30          System.out.println("Special Characters: "+specialCharacters);
31      }
32 }
```

```
It's a pallindrome
```

Q.5 WAP to implement Anagram checking least inbuilt methods being used.

```java
1 package StringsAssignment;
2 import java.util.*;
3 public class ques5 {
4     public static void main(String[] args) {
5          String str1 = "gram";
6          String str2 = "arm";
7          // hash map
8          HashMap<Character,Integer> map = new HashMap<>();
9          // iterate over str1 and put all characters with their freq. in map
10         for(int i=0;i<str1.length();i++) {
11             if(!map.containsKey(str1.charAt(i))) {
12                 map.put(str1.charAt(i), 1);
13             }else {
14                 map.put(str1.charAt(i), map.get(str1.charAt(i))+1);
15             }
16         }
17         // now, iterate over str2
18         for(int i=0;i<str2.length();i++) {
19             if(map.containsKey(str2.charAt(i))) {
20                 map.put(str2.charAt(i),map.get(str2.charAt(i))-1);
21             }else {
22                 System.out.println("Not anagrams");
23                 return;
24             }
25         }
26         // iterate over map
27         for(Map.Entry<Character, Integer> entry:map.entrySet()) {
28             if(entry.getValue()>0) {
29                 System.out.println("Not anagrams");
30                 return;
31             }
32         }
33         System.out.println("Anagrams");
34     }
35 }
```

```
Not anagrams
```

Q.6 WAP to implement Pangram checking least inbuilt methods being used.

```java
package StringsAssignment;
public class ques6 {
    public static void main(String[] args) {

        String str = "Abcdefghijklmnopqrstuvwxyz12";
        // convert to lowercase
        str = str.toLowerCase();

        for(char i='a';i<='z';i++) {
            if(!str.contains(String.valueOf(i))) {
                System.out.println("Not pangram");
                return;
            }
        }
        System.out.println("Pangram");
    }
}
```

```
Pangram
```

Q.7 WAP to find if a string contains all unique characters .

```java
package StringsAssignment;
import java.util.*;
public class ques7 {
    public static void main(String[] args) {
        String str = "Programming in Java";
        // hashmap
        HashMap<Character,Integer> map = new HashMap<>();

        for(int i=0;i<str.length();i++) {
            if(map.containsKey(str.charAt(i))) {
                System.out.println("String does not contain all unique characters")
                return;
            }else {
                map.put(str.charAt(i), 1);
            }
        }
        System.out.println("String contains all unique characters");
    }
}
```

```
String does not contain all unique characters
```

Q.8 WAP to find the maximum occurring character in a string.

```java
package StringsAssignment;
import java.util.*;
public class ques8 {
    public static void main(String[] args) {
        String str ="Programming in Java";
        // hashmap
        HashMap<Character,Integer> map = new HashMap<>();
        for(int i=0;i<str.length();i++) {
            if(!map.containsKey(str.charAt(i))) {
                map.put(str.charAt(i),1);
            }else {
                map.put(str.charAt(i), map.get(str.charAt(i))+1);
            }
        }
        int freq = -1;
        int curFreq = -1;
        for(Map.Entry<Character,Integer> entry:map.entrySet()) {
            curFreq = entry.getValue();
            if(curFreq>freq) {
                freq = curFreq;
            }
        }
        for(int i=0;i<str.length();i++) {
            if(map.get(str.charAt(i))==freq) {
                System.err.println(str.charAt(i));
                return;
            }
        }
    }
}
```

Console ×

<terminated> ques8 (1) [Java Application] C:\Users\dell6\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\javaw.exe (15-Feb-2023, 11:53:40 am – 11:53:40 am) [pid: 2

a