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

**Ans.**

```
import java.util.*;
class Main
{
static String removeDuplicate(char str[], int n)
{
int index = 0;
for (int i = 0; i < n; i++)
{
int j;
for (j = 0; j < i; j++)
{
if (str[i] == str[j])
{
break;
}
}
// For not duplicacy j==i
if (j == i)
{
str[index++] = str[i];
}
}
return String.valueOf(Arrays.copyOf(str, index));
}
public static void main(String[] args)
{
System.out.print("Enter a string: ");
Scanner scan=new Scanner(System.in);
String string= scan.nextLine();
char str[] = string.toCharArray();
int n = str.length;
System.out.println(removeDuplicate(str, n));
}
}
```

2. **WAP to print Duplicates characters from the String.**

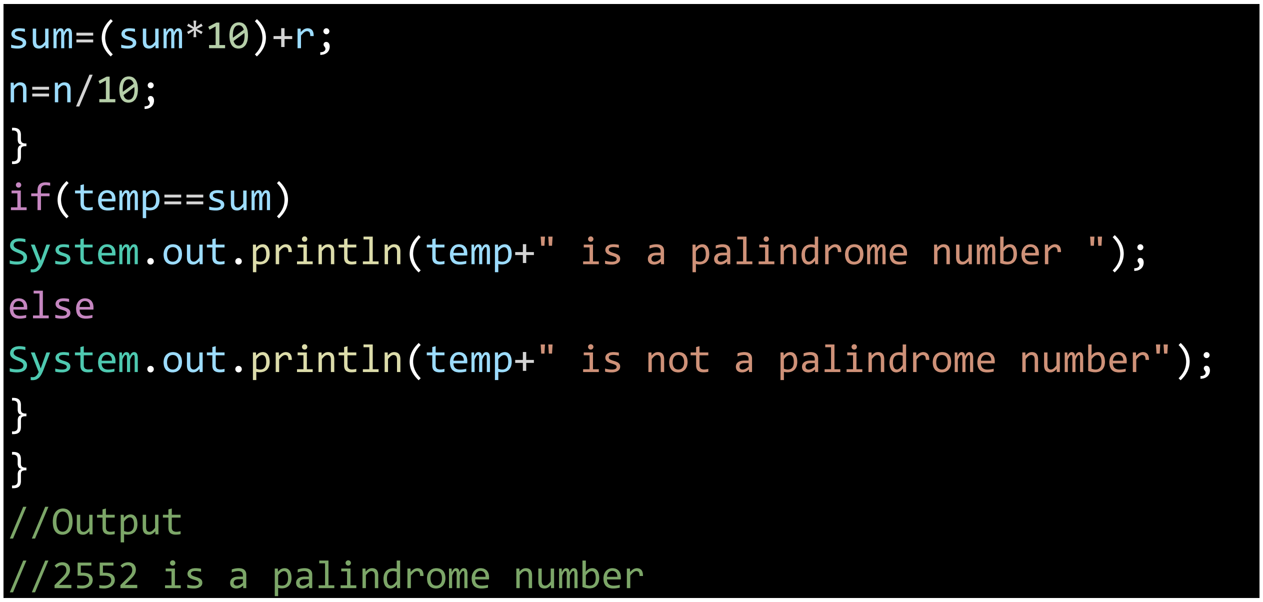
**Ans.**

```
import java.util.*;
class Main
{
static String removeDuplicate(char str[], int n)
{
int index = 0;
for (int i = 0; i < n; i++)
{
int j;
for (j = 0; j < i; j++)
{
if (str[i] == str[j])
{
break;
}
}
// For duplicacy j!=i
if (j != i)
{
str[index++] = str[i];
}
}
return String.valueOf(Arrays.copyOf(str, index));
}
public static void main(String[] args)
{
System.out.print("Enter a string: ");
Scanner scan=new Scanner(System.in);
String string= scan.nextLine();
char str[] = string.toCharArray();
int n = str.length;
System.out.println(removeDuplicate(str, n));
}
}
```

**3. WAP to check if "2552" is palindrome or not.**

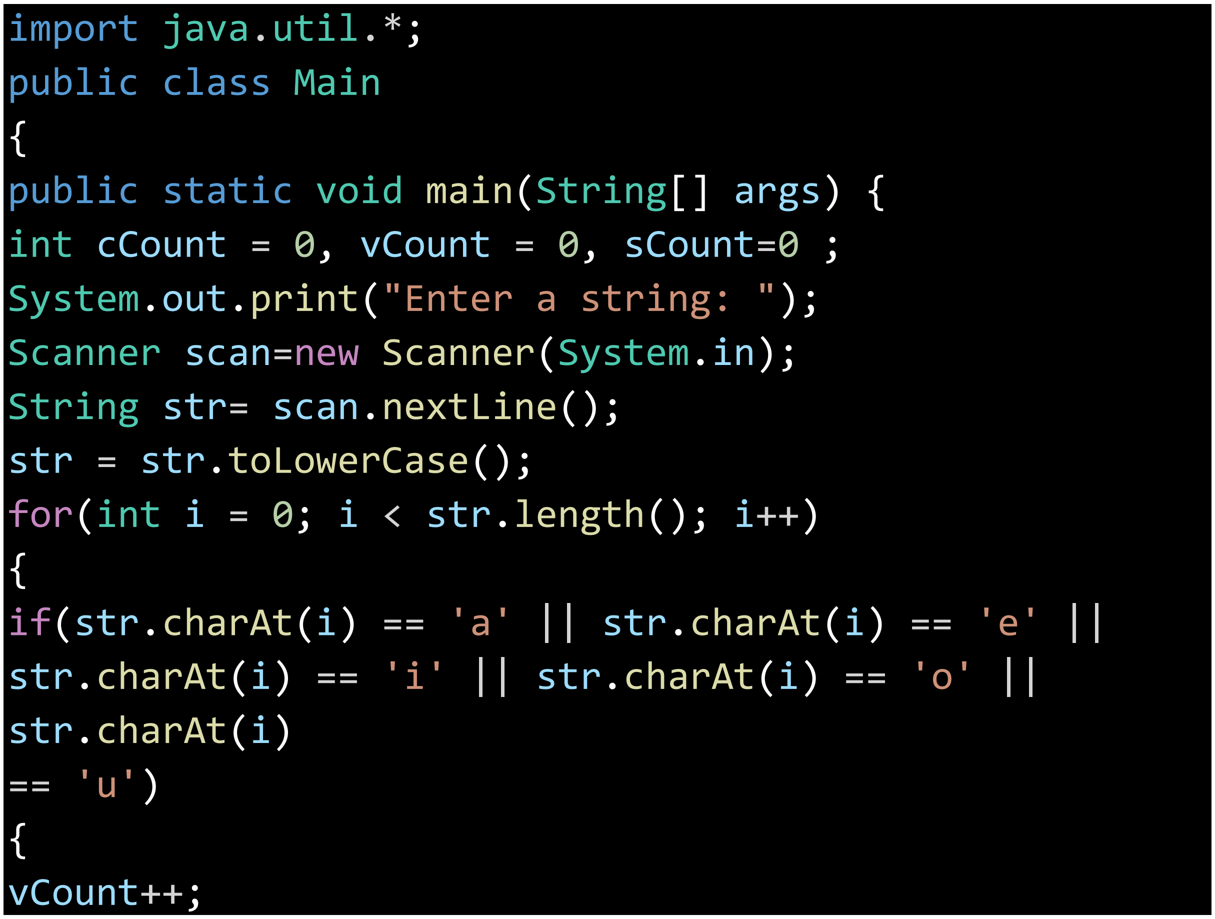
**Ans.**

```
import java.util.*;
class Main
{
public static void main(String args[])
{
int r,sum=0,temp;
System.out.print("Enter number: ");
Scanner scan=new Scanner(System.in);
int n= scan.nextInt();
temp=n;
while(n>0)
{
r=n%10;
```



```java
sum=(sum*10)+r;
n=n/10;
}
if(temp==sum)
System.out.println(temp+" is a palindrome number ");
else
System.out.println(temp+" is not a palindrome number");
}
}
//Output
//2552 is a palindrome number
```

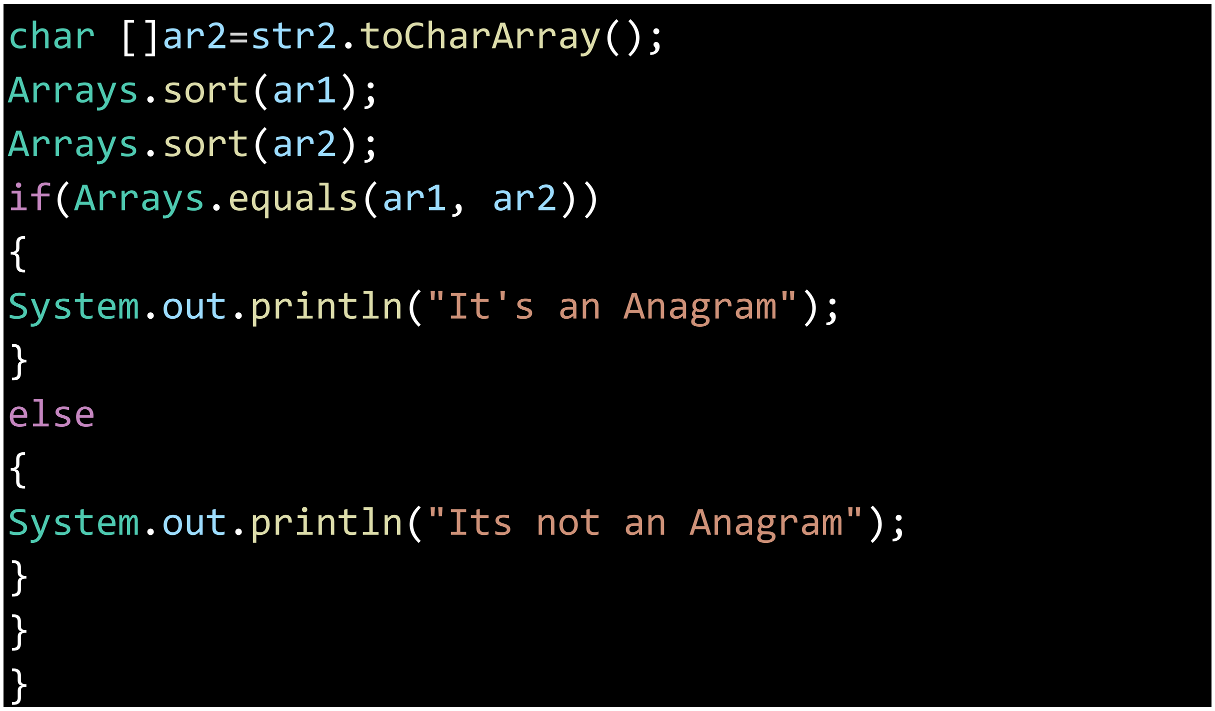
4. **WAP to count the number of consonants, vowels & special characters in a String .**



```java
import java.util.*;
public class Main
{
public static void main(String[] args) {
int cCount = 0, vCount = 0, sCount=0 ;
System.out.print("Enter a string: ");
Scanner scan=new Scanner(System.in);
String str= scan.nextLine();
str = str.toLowerCase();
for(int i = 0; i < str.length(); i++)
{
if(str.charAt(i) == 'a' || str.charAt(i) == 'e' ||
str.charAt(i) == 'i' || str.charAt(i) == 'o' ||
str.charAt(i)
== 'u')
{
vCount++;
}
else if(str.charAt(i) >= 'a' && str.charAt(i)<='z')
{
cCount++;
}
else if(!Character.isDigit(str.charAt(i)))
{
sCount++;
}
}
System.out.println("Number of consonants: " + cCount);
System.out.println("Number of vowels: " + vCount);
System.out.println("Number of special character: " +
sCount);
}
}
```

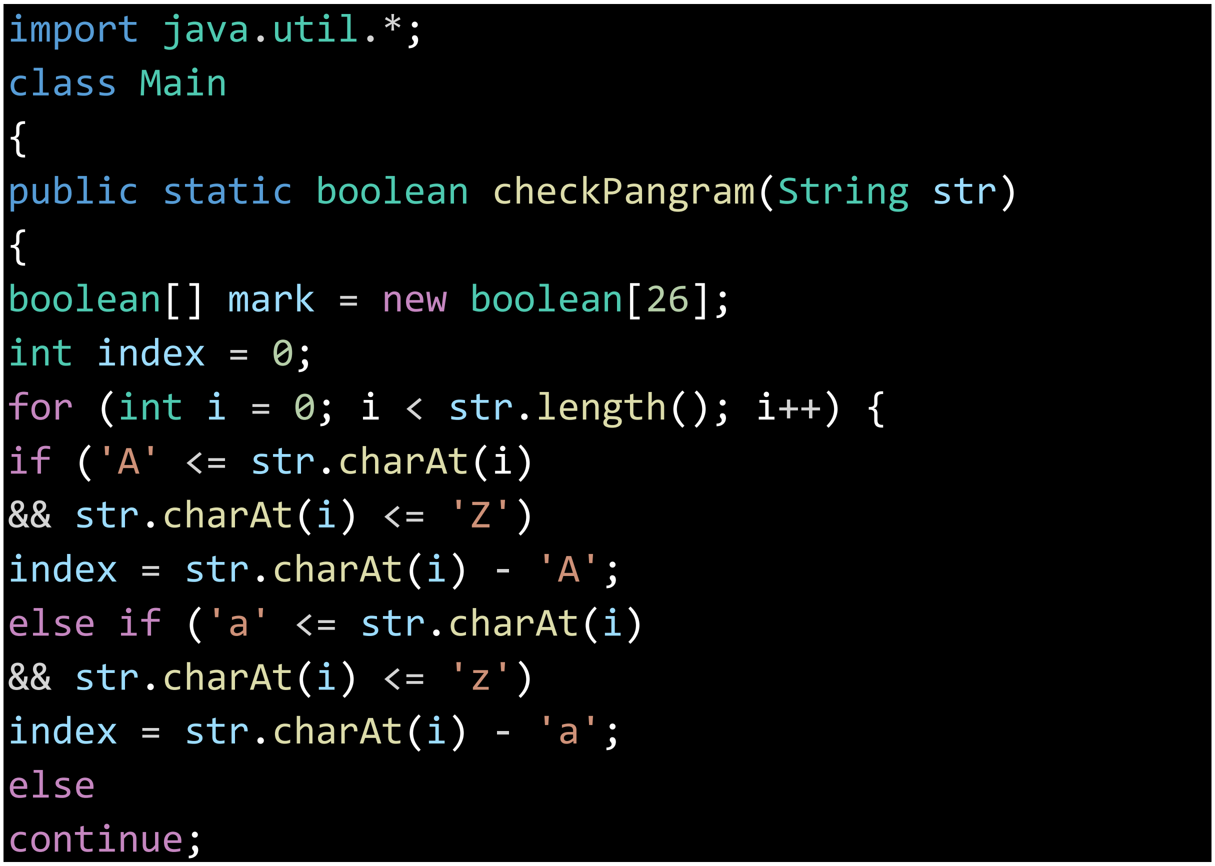
**5. WAP to implement Anagram Checking least inbuilt methods being used.**

```
import java.util.*;
public class Main
{
public static void main(String[] args)
{
Scanner scan=new Scanner(System.in);
System.out.print("Enter 1st string: ");
String str1= scan.nextLine();
System.out.print("Enter 2nd string: ");
String str2= scan.nextLine();
str1=str1.toLowerCase();
str2=str2.toLowerCase();
char []ar1=str1.toCharArray();
```

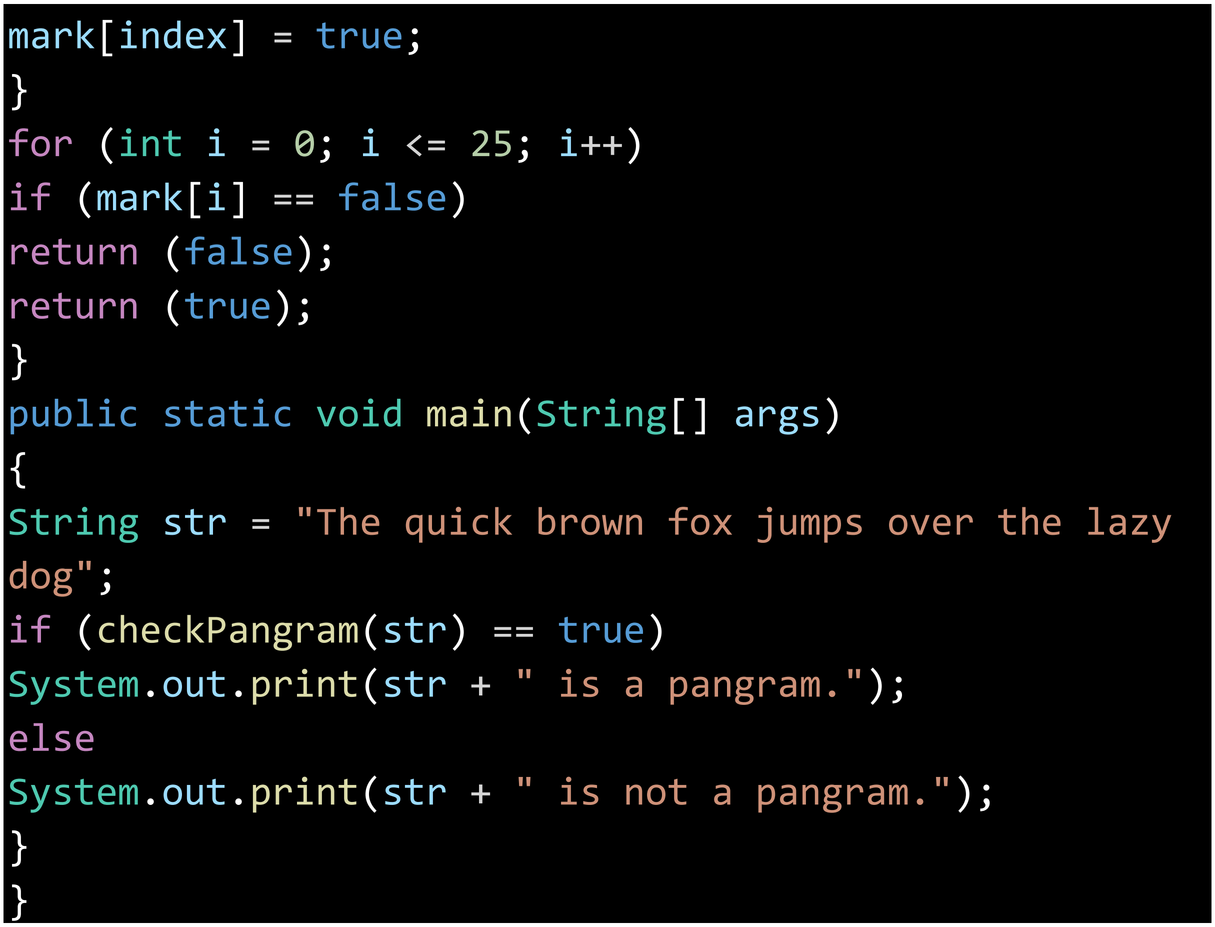


```
char []ar2=str2.toCharArray();
Arrays.sort(ar1);
Arrays.sort(ar2);
if(Arrays.equals(ar1, ar2))
{
System.out.println("It's an Anagram");
}
else
{
System.out.println("Its not an Anagram");
}
}
}
```

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

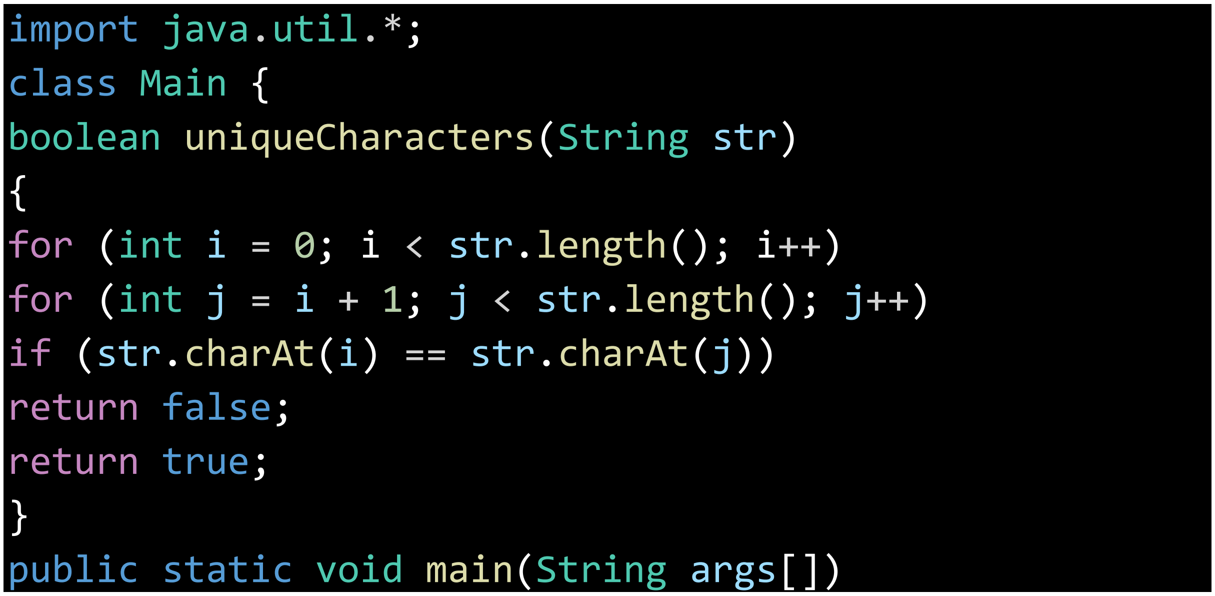


```
import java.util.*;
class Main
{
public static boolean checkPangram(String str)
{
boolean[] mark = new boolean[26];
int index = 0;
for (int i = 0; i < str.length(); i++) {
if ('A' <= str.charAt(i)
&& str.charAt(i) <= 'Z')
index = str.charAt(i) - 'A';
else if ('a' <= str.charAt(i)
&& str.charAt(i) <= 'z')
index = str.charAt(i) - 'a';
else
continue;
```



```
mark[index] = true;
}
for (int i = 0; i <= 25; i++)
if (mark[i] == false)
return (false);
return (true);
}
public static void main(String[] args)
{
String str = "The quick brown fox jumps over the lazy 
dog";
if (checkPangram(str) == true)
System.out.print(str + " is a pangram.");
else
System.out.print(str + " is not a pangram.");
}
}
```

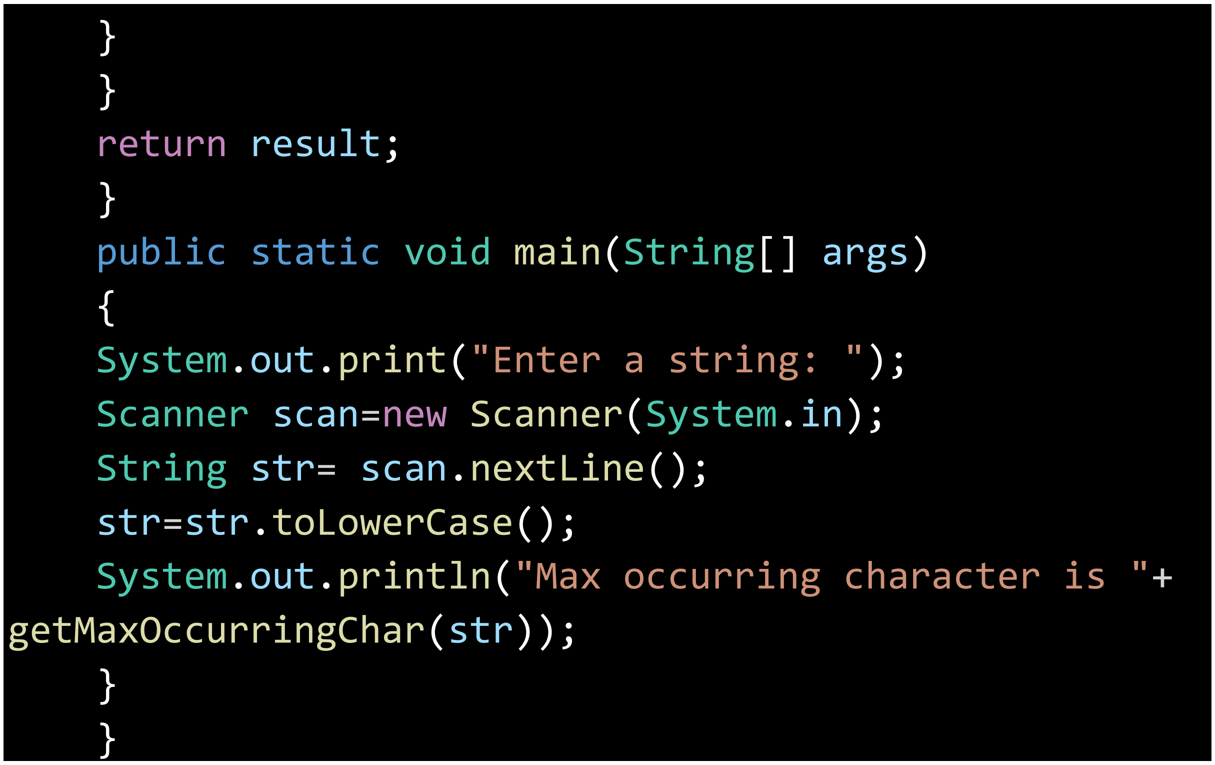
7. **WAP to find if String contains all unique characters.**



```
import java.util.*;
class Main {
boolean uniqueCharacters(String str)
{
for (int i = 0; i < str.length(); i++)
for (int j = i + 1; j < str.length(); j++)
if (str.charAt(i) == str.charAt(j))
return false;
return true;
}
public static void main(String args[])
{
Main obj = new Main();
String input = "PwskiLl";
input=input.toLowerCase();
if (obj.uniqueCharacters(input))
System.out.println("The String " + input + " has all
unique characters");
else
System.out.println("The String " + input + " has
duplicate characters");
}
}
```

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

```
import java.util.*;
public class Main
{
static final int ASCII_SIZE = 256;
static char getMaxOccurringChar(String str)
{
    int count[] = new int[ASCII_SIZE];
    int len = str.length();
    for (int i = 0; i < len; i++)
    count[str.charAt(i)]++;
    int max = -1;
    char result = ' ';
    for (int i = 0; i < len; i++) {
    if (max < count[str.charAt(i)]) {
    max = count[str.charAt(i)];
    result = str.charAt(i);
```



```
    }
    }
    return result;
    }
    public static void main(String[] args)
    {
    System.out.print("Enter a string: ");
    Scanner scan=new Scanner(System.in);
    String str= scan.nextLine();
    str=str.toLowerCase();
    System.out.println("Max occurring character is "+
getMaxOccurringChar(str));
    }
    }
```