**Q1) WAP to remove Duplicates from a String. (Take any String ex with duplicates character)**

**import** java.util.\*;

**import** java.io. \*;

**class** RemoveDuplicates{

**static** **void** removeDuplicate(**char** str[], **int** length) {

**int** index = 0;

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

**for** (int j = 0; j < i; j++){

**if** (str[i] == str[j])

**break**;

}

**if** (j == i)

str[index++] = str[i];

}

System.*out*.println("String with duplicates");

System.*out*.println(String.*valueOf*(Arrays.*copyOf*(str, index)));

}

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

String info;

System.*out*.println("Enter String with duplicate letters");

Scanner sc=**new** Scanner(System.*in*);

info=sc.next();

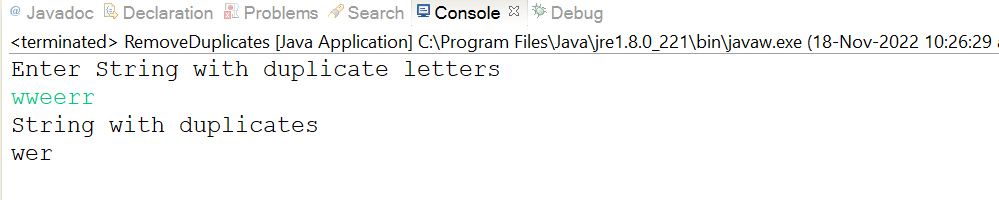
**char** str[] = info.toCharArray();

**int** len = str.length;

*removeDuplicate*(str, len);

}

}



**Q2) WAP to print Duplicates characters from the String.**

**import** java.util.Scanner;

**public** **class** PrintDuplicates {

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

String string1;

System.*out*.println("Enter String with duplicate letters");

Scanner sc=**new** Scanner(System.*in*);

string1=sc.next();

**int** count;

**char** string[] = string1.toCharArray();

System.*out*.println("Duplicate characters in a given string: ");

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

count = 1;

**for**(**int** j = i+1; j <string.length; j++) {

**if**(string[i] == string[j] && string[i] != ' ') {

count++;

string[j] = '0';

}

}

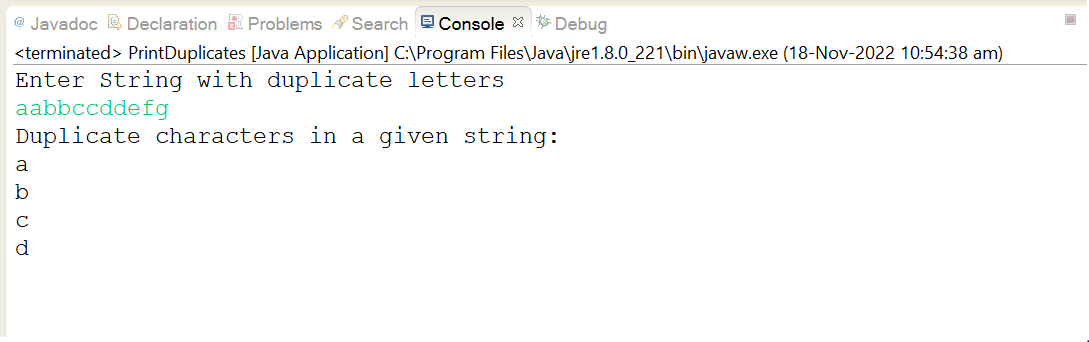
**if**(count > 1 && string[i] != '0')

System.*out*.println(string[i]);

}

}

}



**Q3)** WAP to check if “2552” is palindrome or not.

**public** **class** Palindrome {

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

String string = "2552";

**boolean** flag = **true**;

string = string.toLowerCase();

**for**(**int** i = 0; i < string.length()/2; i++){

**if**(string.charAt(i) != string.charAt(string.length()-i-1)){

flag = **false**;

**break**;

}

}

**if**(flag)

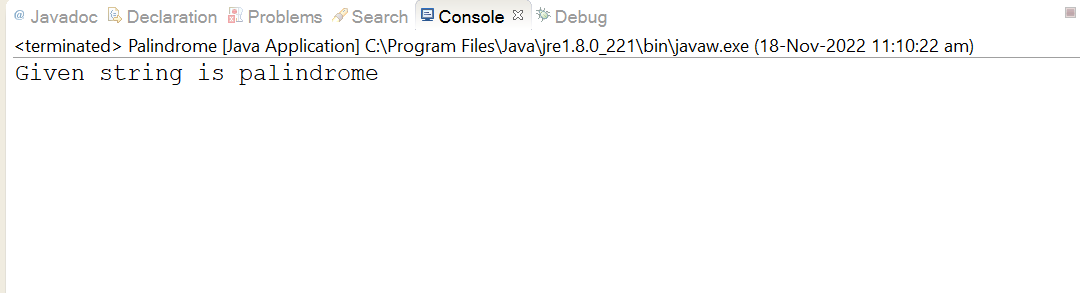
System.*out*.println("Given string is palindrome");

**else**

System.*out*.println("Given string is not a palindrome");

}

}



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

**import** java.io.\*;

**import** java.util.Scanner;

**public** **class** Count{

**static** **void** countCharacterType(String str){

**int** vowels = 0, consonant = 0, specialChar = 0,digit = 0;

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

**char** ch = str.charAt(i);

**if** ( (ch >= 'a' && ch <= 'z') ||(ch >= 'A' && ch <= 'Z') ) {

ch = Character.*toLowerCase*(ch);

**if** (ch == 'a' || ch == 'e' || ch == 'i' ||ch == 'o' || ch == 'u')

vowels++;

**else**

consonant++;

}

**else** **if** (ch >= '0' && ch <= '9')

digit++;

**else**

specialChar++;

}

System.*out*.println("Vowels: " + vowels);

System.*out*.println("Consonant: " + consonant);

System.*out*.println("Digit: " + digit);

System.*out*.println("Special Character: " + specialChar);

}

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

String str;

System.*out*.println("Enter the string");

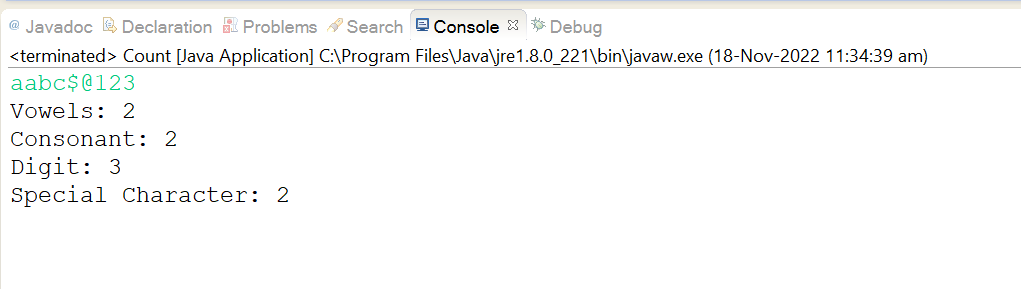
Scanner sc=**new** Scanner(System.*in*);

str=sc.next();

*countCharacterType*(str);

}

}



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

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** AnagramChecker{

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

Scanner scanner = **new** Scanner(System.*in*);

System.*out*.print("Enter the First String : ");

String s1 = scanner.nextLine();

System.*out*.print("Enter the second String : ");

String s2 = scanner.nextLine();

**if**(*checkAnagram*(s1, s2))

System.*out*.println(s1+" and "+s2+" are Anagrams");

**else**

System.*out*.println(s1+" and "+s2+" are NOT Anagrams");

}

**public** **static** **boolean** checkAnagram(String s1, String s2) {

s1 = s1.replaceAll("\\s", "");

s2 = s2.replaceAll("\\s", "");

**if**(s1.length() != s2.length())

**return** **false**;

**else**{

**char**[] arr1 = s1.toLowerCase().toCharArray();

**char**[] arr2 = s2.toLowerCase().toCharArray();

Arrays.*sort*(arr1);

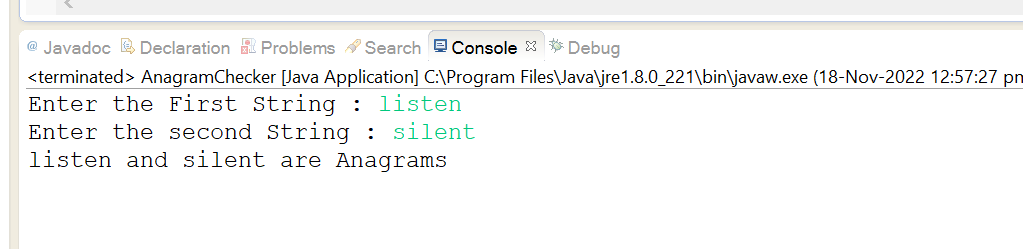
Arrays.*sort*(arr2);

**return** (Arrays.*equals*(arr1, arr2));

}

}

}



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

**import** java.util.Scanner;

**public** **class** PanagramStr {

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

Scanner sc=**new** Scanner(System.*in*);

System.*out*.println("Enter Your String:");

String str=sc.nextLine();

str=str.replaceAll("","").toLowerCase();

String s="";

**for**(**char** i='a';i<='z';i++){

**if**(str.indexOf(i)!=-1){

s=s+i;// empty string+character

}

}

**if**(s.length()==26){

System.*out*.println("Pangram");

}

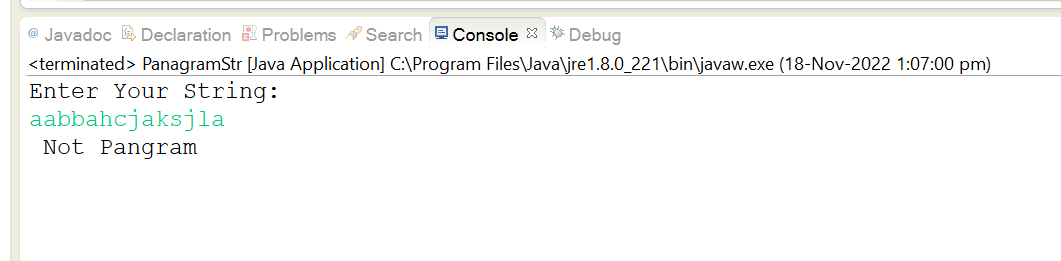
**else**{

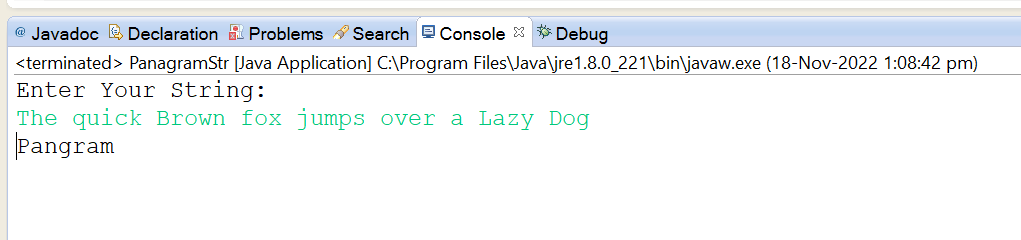
System.*out*.println(" Not Pangram");

}

}

}





**Q.7) WAP to implement Pangram Checking with least inbuilt methods being used.**

**import** java.util.\*;

**class** Uniquechar{

**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[]){

Uniquechar obj = **new** Uniquechar();

String input = "Hello";

**if** (obj.uniqueCharacters(input))

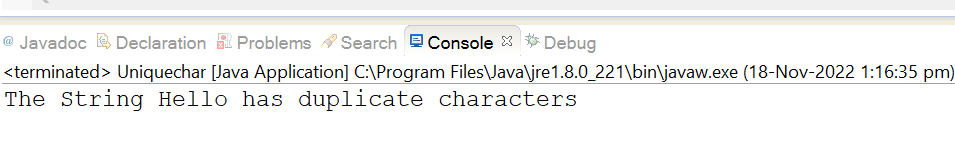
System.*out*.println("The String " + input + " has all unique characters");

**else**

System.*out*.println("The String " + input + " has duplicate characters");

}

}



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

**public** **class** MaxOccur{

**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; // Initialize max count

**char** result = ' '; // Initialize 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){

String str = "sample string";

System.*out*.println("Max occurring character is "+ *getMaxOccurringChar*(str));

}

}

