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

Program:-

import java.util.\*;

class HelloWorld {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String n=s.next();

String r="";//String.valueOf(n.charAt(0));

int c=0;

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

{

int j=0;

for(j=0;j<i;j++)

{

if(n.charAt(i)==n.charAt(j))

{

break;

}

}if(i==j)

r=r+n.charAt(i);

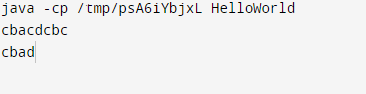
}

System.out.println(r);

}

}

Output:-



2. WAP to print Duplicates characters from the String.

Program:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String string3 = s.nextLine();

String string1=string3.toLowerCase();

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')

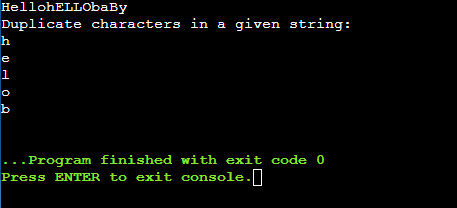
System.out.println(string[i]);

}

}

}

OutPut:-



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

Progranm:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String s1 = s.nextLine();

String t="";

for(int i=s1.length()-1;i>=0;i--)

{

t=t+s1.charAt(i);

}

if(s1.equals(t))

{

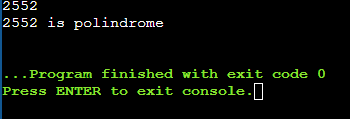
System.out.println(s1 +" is polindrome");

}

}

}

Output:-



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

Program:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String s1 = s.nextLine();

String s2=s1.toLowerCase();

int v=0,c=0,o=0;

for(int i=0;i<s1.length();i++)

{

if(s2.charAt(i)=='a'||s2.charAt(i)=='e'||s2.charAt(i)=='i'||s2.charAt(i)=='o'||s2.charAt(i)=='u')

{

v++;

}

else if(((s2.charAt(i)>'a')&&(s2.charAt(i)<'z'))&&(s2.charAt(i)!='a'&&s2.charAt(i)!='e'&&s2.charAt(i)!='i'&&s2.charAt(i)!='o'&&s2.charAt(i)!='u'))

{

c++;

}

else

{

o++;

}

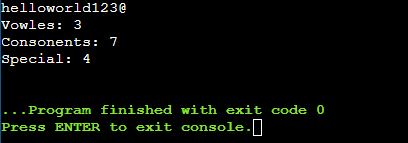
}

System.out.println("Vowles: "+v+"\n"+"Consonents: "+c+"\n"+"Special: "+o);

}

}

Output:-



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

Program:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String s1 = s.nextLine();

String s2=s.nextLine();

s1=s1.toLowerCase();

s2=s2.toLowerCase();

int l1=s1.length();

int l2=s2.length();

int c1=0,c2=0;

if(l1==l2)

{

for(int i=0;i<s1.length();i++)

{

String t=String.valueOf(s1.charAt(i));

for(int j=i+1;j<s1.length();j++)

{

if(t.equals(s1.charAt(j)))

{

c1++;

}

if(t.equals(s2.charAt(j)))

{

c2++;

}

}

if(c1==c2)

{

continue;

}

else{

System.out.println("not anagram");

}

}

System.out.println("Anagram");

}

else

{

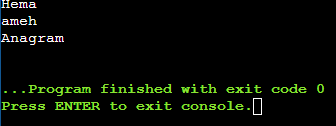
System.out.println("Not an anagram");

}

}

}

Output:-



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

Program:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String str = s.nextLine();

boolean[] alphaList = new boolean[26];

int index = 0;

int flag = 1;

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

if ( str.charAt(i) >= 'A' && str.charAt(i) <= 'Z') {

index = str.charAt(i) - 'A';

}else if( str.charAt(i) >= 'a' && str.charAt(i) <= 'z') {

index = str.charAt(i) - 'a';

}

alphaList[index] = true;

}

for (int i = 0; i <= 25; i++) {

if (alphaList[i] == false)

flag = 0;

}

System.out.print("String: " + str);

if (flag == 1)

System.out.print("\nThe above string is a pangram.");

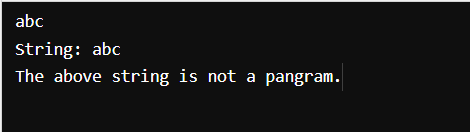
else

System.out.print("\nThe above string is not a pangram.");

}

}

Output:-



Text

Description automatically generated

7. WAP to find if String contains all unique characters

Program:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String str = s.nextLine();

int l=str.length();

int c=0;

for(int i=0;i<l;i++)

{

//String d=String.valueOf(str.charAt(i));

for(int j=i+1;j<l;j++)

{

if(str.charAt(i)==str.charAt(j))

{

c++;

}

}

}

if(c!=0)

{

System.out.println("String doesn't have unique characters");

}

else{

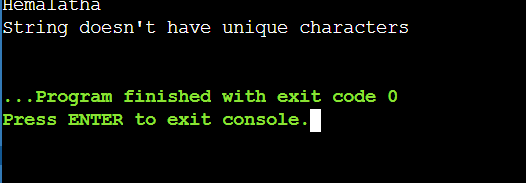
System.out.println("String is unique");

}

}

}

Output:-



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

Program:-

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String str = s.nextLine();

int l=str.length(),r=0,c=0;

String t="";

for(int i=0;i<l;i++)

{

for(int j=i+1;j<l;j++)

{

if(str.charAt(i)==str.charAt(j))

{

c++;

}

}

if(c>r)

{

r=c;

t=String.valueOf(str.charAt(i));

}

}

System.out.println("maximum occurring character in a String is: "+t);

}

}

Output:-

