1. **WAP to reverse a given string**

import java.util.\*;

class ReverseString

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string to be reversed: ");

String str = sc.nextLine();

String rev = "";

int leng = str.length();

for(int i=leng-1 ; i>=0 ; i--)

{

rev = rev + str.charAt(i);

}

System.out.println("\nThe reversed string is: "+rev);

}

}

1. **WAP to check is a given string is palindrome or not**

import java.util.\*;

class PalindromeString

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string to be reversed: ");

String str = sc.nextLine();

String rev = "";

int leng = str.length();

for(int i=leng-1 ; i>=0 ; i--)

{

rev = rev + str.charAt(i);

}

if(str.equals(rev))

{

System.out.println("\nThe given string is palindrome!");

}

else

{

System.out.println("\nThe given string is not palindrome!");

}

}

}

1. **WAP to find the maximum and minimum occurring character in a given string**

import java.util.\*;

class PalindromeString

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string to be reversed: ");

String str = sc.nextLine();

String rev = "";

int leng = str.length();

for(int i=leng-1 ; i>=0 ; i--)

{

rev = rev + str.charAt(i);

}

if(str.equals(rev))

{

System.out.println("\nThe given string is palindrome!");

}

else

{

System.out.println("\nThe given string is not palindrome!");

}

}

}

1. **WAP to count a number of words in string**

import java.util.\*;

class CountWords

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

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

String str = sc.nextLine();

String strarr[] = str.split(" ");

int count = strarr.length;

System.out.println("\nThe number of words in the given string is: "+count);

}

}

1. **WAP to print all permutation of strings**

import java.util.\*;

public class StringPermutation

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

String s;

String result = "";

System.out.print("Enter a String: ");

s = sc.nextLine();

System.out.println("All permutation of this string: ");

Permutation(s,result);

}

private static void Permutation(String string, String answer)

{

if (string.length() == 0)

{

System.out.print(answer + " ");

return;

}

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

{

char c = string.charAt(i);

String left\_subString = string.substring(0,i);

String right\_subString = string.substring(i+1);

String rest = left\_subString + right\_subString;

Permutation(rest, answer + c);

}

}

}

1. **WAP to print the duplicate characters from the given string**

import java.util.\*;

public class DuplicateCharacters

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enthe the string: ");

String string1 = sc.nextLine();

int count=1;

//Converts given string into character array

char string[] = string1.toCharArray();

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

//Counts each character present in the string

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

{

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

{

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

{

count++;

//Set string[j] to 0 to avoid printing visited character

string[j] = '0';

}

}

//A character is considered as duplicate if count is greater than 1

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

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

}

}

}

1. **WAP to remove a given character from string**

import java.util.Scanner;

public class RemoveCharacter

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string: ");

String str = scanner.nextLine();

// Get the character to be removed from the user

System.out.println("Enter the character to be removed: ");

char ch = scanner.next().charAt(0);

// Remove the character from the string

String newStr = str.replace(ch, '');

System.out.println("The new string is: " + newStr);

}

}

1. **WAP to remove all duplicates from the string**

import java.util.\*;

public class RemoveDuplicates

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

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

String str1 = sc.nextLine();

System.out.println("The given string is: " + str1);

System.out.println("After removing duplicates characters the new string is: " + removeDuplicateChars(str1));

}

private static String removeDuplicateChars(String sourceStr) {

// Convert the input string to a character array.

char[] arr1 = sourceStr.toCharArray();

// Initialize an empty string to store the resulting string without duplicates.

String targetStr = "";

// Loop through each character in the character array.

for (char value: arr1)

{

// Check if the character doesn't exist in the targetStr.

if (targetStr.indexOf(value) == -1)

{

// If the character doesn't exist, add it to the targetStr.

targetStr += value;

}

}

// Return the resulting string without duplicates.

return targetStr;

}

}

1. **WAP to remove characters from the first string which are present in the second string**

import java.util.Scanner;

public class RemoveCharFromFirst {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the first string: ");

String str1 = sc.nextLine();

System.out.println("Enter the second string: ");

String str2 = sc.nextLine();

String result = "";

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

// Check if the character is present in the second string.

if (!str2.contains(str1.charAt(i))) {

// Add the character to the result string.

result += str1.charAt(i);

}

}

System.out.println("The resulting string is: " + result);

}

}

1. **WAP to find the first non-repeating character in a given string**

import java.util.\*;

class NonRepeatingChar

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter a String:");

String word = sc.nextLine();

boolean flag = true;

for(char i :word.toCharArray())

{

if (word.indexOf(i) == word.lastIndexOf(i))

{

System.out.println("The first character which is not repeating is: "+ i);

flag = false;

break;

}

}

if(flag== true)

{

System.out.println("There is no non-repeating character in the input string");

}

}

}

1. **WAP to check if two given string is the anagram of each other**

import java.util.\*;

public class AnagramString

{

public static void main (String [] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter first string: ");

String str1 = sc.nextLine();

System.out.println("Enter second string: ");

String str2 = sc.nextLine();

str1 = str1.toLowerCase();

str2 = str2.toLowerCase();

if (str1.length() != str2.length())

{

System.out.println("Both the strings are not anagram");

}

else

{

//Converting both the arrays to character array

char[] string1 = str1.toCharArray();

char[] string2 = str2.toCharArray();

//Sorting the arrays using in-built function sort ()

Arrays.sort(string1);

Arrays.sort(string2);

//Comparing both the arrays using in-built function equals ()

if(Arrays.equals(string1, string2) == true)

{

System.out.println("Both the strings are anagram");

}

else

{

System.out.println("Both the strings are not anagram");

}

}

}

}