### how to reverse a string in java without using any API.

**Reversing string by converting into string array and iterating it reversely**

Here we have created array of string, we are going to last string in first forloop and in second for loop we are printing from last index reverse order

class reverseWithoutAPI   
{   
**public** **static** **void** main(String a[]) {

String str = "Java 2 career";

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

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

**for** (**int** i = arr.length - 1; i >= 0; i--) {

**char** temp[] = arr[i].toCharArray();

**for** (**int** j = temp.length - 1; j >= 0; j--) {

System.***out***.print(temp[j]);

}

System.***out***.print("");

}

}  
}

**Reverse with swapping with third variable**

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

String str = "12345";

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

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

**char** temp = arr[i];

arr[i] = arr[arr.length - i-1];

arr[arr.length - i-1] = temp;

}

String reversedSstring = **new** String(arr);

System.***out***.println(reversedSstring);

}

output-Reversed String of Java 2 career:reerac 2 avaJ

### how to find duplicate characters(no of occurrence) in a string in java. Or how to count occurrence of each character in a string in java.

**without using predefined method or collection**

### how do you remove all white spaces from a string in java.

public class RemoveAllSpace {

    public static void main(String[] args) {

        String str = "India     Is My    Country";

***//1st way***

        String noSpaceStr = str.replaceAll("\\s", ""); // using built in method

        System.out.println(noSpaceStr);

**//2nd way**

        char[] strArray = str.toCharArray();

        StringBuffer stringBuffer = new StringBuffer();

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

            if ((strArray[i] != ' ') && (strArray[i] != '\t')) {

                stringBuffer.append(strArray[i]);

            }

        }

        String noSpaceStr2 = stringBuffer.toString();

        System.out.println(noSpaceStr2);

    }

}

Output:

IndiaIsMyCountry

IndiaIsMyCountry

### 4.> how to check given String is palindrome or not

package com.includehelp.stringsample;

import java.util.Scanner;

/\*\*

 \* Easiest way to check Given String is Palindrome String or not

 \*@authorincludehelp

 \*/

publicclass PalindromString {

staticboolean isPalindromString(String inputStr){

StringBuilder sb =new StringBuilder(inputStr);

String reverseStr = sb.reverse().toString();

return(inputStr.equalsIgnoreCase(reverseStr));

}

publicstaticvoid main(String[] args){

Scanner sc =new Scanner(System.in);

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

String inString = sc.next();

if(isPalindromString(inString)){

System.out.println(inString +" is a Palindrom String");

}

else{

System.out.println(inString +" is not a Palindrom String");

}

}

}

Output

First run:

Enter String : india

india is not a Palindrom String

Second run:

Enter String : abcba

abcba is a Palindrom String

### 5.> how to check if two String are anargms.

|  |
| --- |
| // JAVA program to check whether two strings  // are anagrams of each other  importjava.io.\*;  importjava.util.Arrays;  importjava.util.Collections;    classGFG {        /\* function to check whether two strings are      anagram of each other \*/      staticbooleanareAnagram(char[] str1, char[] str2)      {          // Get lenghts of both strings          intn1 = str1.length;          intn2 = str2.length;            // If length of both strings is not same,          // then they cannot be anagram          if(n1 != n2)              returnfalse;            // Sort both strings          Arrays.sort(str1);          Arrays.sort(str2);            // Compare sorted strings          for(inti = 0; i < n1; i++)              if(str1[i] != str2[i])                  returnfalse;            returntrue;      }        /\* Driver program to test to print printDups\*/      publicstaticvoidmain(String args[])      {          charstr1[] = { 't', 'e', 's', 't'};          charstr2[] = { 't', 't', 'e', 'w'};          if(areAnagram(str1, str2))              System.out.println("The two strings are"                                 + " anagram of each other");          else              System.out.println("The two strings are not"                                 + " anagram of each other");      }  }    // This code is contributed by Nikita Tiwari. |

Output:

The two strings are not anagram of each other

6.> how to find duplicate character in a String.

publicclassDuplStr{

publicstaticvoid main(String argu[]){

String str ="w3schools";

int cnt =0;

char[] inp = str.toCharArray();

System.out.println("Duplicate Characters are:");

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

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

if(inp[i]== inp[j]){

System.out.println(inp[j]);

cnt++;

break;

}

}

}

}

}

Program Output:

Duplicate Characters are: s o

***STRING PROGRAMS***

1. [Java program to remove duplicate characters in String](#_Java_program_to_6)
2. [Java Program to count the total number of vowels and consonants in a string](#_Java_Program_to_7)
3. **Reverse words in the string**
4. **Java Program to determine whether two strings are the anagram**
5. **Java Program to determine whether two strings are the Pangram**
6. **Java Program to divide a string in 'N' equal parts.**
7. **Java Program to find all subsets of a string**
8. **Java Program to find the longest repeating sequence in a string**
9. **Java Program to find all the permutations of a string**
10. **Java Program to remove all the white spaces from a string**
11. **Java Program to replace lower-case characters with upper-case and vice-versa**
12. **Java Program to replace the spaces of a string with a specific character**
13. **Java Program to determine whether a given string is palindrome**
14. **Java Program to determine whether one string is a rotation of another**
15. **Java Program to find maximum and minimum occurring character in a string**
16. **Java Program to find Reverse of the string**
17. **Java program to find the duplicate characters in a string**
18. **Java program to find the duplicate words in a string**
19. **Java Program to find the frequency of characters**
20. **Java Program to find the largest and smallest word in a string**

#### **Java program to remove duplicate characters in String**

We can remove duplicates from string using following ways.

1. By using for loop and comparing with it’s previous elements
2. By using sorting
3. By using hashing i. e. hashmap
4. Using predefined methods
5. **By using for loop and comparing with it’s previous elements**

public class Test01\_StringReverse {  
 public static void main(String[] args) {  
 String str = "madam";  
 char[] arr = str.toCharArray ();  
 int n = str.length ();  
  
 int index = 0;  
  
 for (int i = 0; i < n; i++) {  
 int k = 0;  
 for (k = 0; k < i; k++) {  
 if (arr[i] == arr[k]) break;  
 }  
 if (i == k) {  
 arr[index++] = arr[i];  
 }  
 }  
 String newStr = new String (arr);  
*// System.out.println(String.valueOf(Arrays.copyOf(arr, index)));* for (int i = 0; i < index; i++) {  
 System.*out*.print (arr[i] + " ");  
 }  
 }  
}

1. **By using sorting**
2. public class Test02\_StringReverseUsingSorting {  
    public static void main(String[] args) {  
    String str = "malayalama";  
    char[] arr = str.toCharArray ();  
    Arrays.*sort*(arr);  
      
    int index1=1,index2=1;  
    int length=arr.length;  
      
    while(index1!=length) {  
    if(arr[index1]!=arr[index1-1]) {  
    arr[index2]=arr[index1];  
    index2++;  
    }  
    index1++;  
    }  
      
    for (int i = 0; i < index2; i++) {  
    System.*out*.print(arr[i]+" ");  
    }  
     
    }  
   }
3. **By using hashing i. e. hashmap**

public class Test03\_StringReverseUsingHashMap {  
  
 public static void main(String[] args) {  
 String str = "madam";  
 char[] arr = str.toCharArray ();  
 int n = str.length ();  
 LinkedHashSet<Character> set = new LinkedHashSet<>();  
  
 for (int i = 0; i < arr.length; i++) {  
 set.add (arr[i]);  
 }  
  
 System.*out*.println (set);  
 }  
}

1. **Using predefined methods**

public class Test04\_StringReverseUsingPredefinedMethods {  
 public static void main(String[] args) {  
 *//Create an empty string* String newstr = "";  
  
 String str = "malayalama";  
 int length = str.length ();  
  
 *// Traverse the string and check for the repeated characters* for (int i = 0; i < length; i++) {  
 *// store the character available at ith index in the string* char charAtPosition = str.charAt (i);  
  
 *// check the index of the charAtPosition. If the indexOf() method returns true add it to the resuting string* if (newstr.indexOf (charAtPosition) < 0) {  
 newstr += charAtPosition;  
 }  
 }  
 *//Print string after removing duplicate characters* System.*out*.println (newstr);  
 }  
}

#### **Java Program to count the total number of vowels and consonants in a string**

public class Test05\_countVowelsConsonents {  
 public static void main(String[] args) {  
 String str = "This is a really simple sentence";  
 String uniqueString = *getUniqueString* (str);  
 int conCount = 0, vcount = 0;  
 char[] arr = uniqueString.toCharArray ();  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] == 'a' || arr[i] == 'i' || arr[i] == 'e' || arr[i] == 'u' || arr[i] == 'o')  
 conCount++;  
 else  
 vcount++;  
 }  
  
 System.*out*.println ("consonent: " + conCount + " Vowels:" + vcount);  
 }  
  
 private static String getUniqueString(String str) {  
 int index = 0, j = 0;  
 char[] arr = str.toCharArray ();  
 int n = arr.length;  
 for (int i = 0; i < n; i++) {  
 for (j = 0; j < i; j++) {  
 if (arr[i] == arr[j]) {  
 break;  
 }  
 }  
 if (i == j) {  
 arr[index++] = arr[i];  
 }  
 }  
 String newString = "".concat (new String (arr));  
 return newString.substring (0, index);  
 }  
}

#### **Reverse words in string**

**package** com.bs.logics;  
  
**public class** Rough {  
 **public static void** main(String[] args) {  
 String str = **"I am an Indian"**;  
 **int** length = str.length ();  
 **char**[] ch = str.toCharArray ();  
 String res=**""**;  
 **for** (**int** i = 0; i < length; i++) {  
 **int** k=i;  
 *//loop to increase i upto white space* **while** (i < length && ch[i] != **' '**) {  
 i++;  
 }  
  
 *//take pointer to before space character* **int** j=i-1;  
 *//append char from initial index to before space char and concate to string* **while**(k<=j){  
 res=res+ch[j];  
 j--;  
 }  
 res=res+**' '**;  
 }  
  
 System.***out***.println (res);  
 }  
}

#### **Java Program to determine whether two strings are the anagram**

#### **Java Program to divide a string in 'N' equal parts.**

#### **Java Program to find all subsets of a string**

#### **Java Program to find the longest repeating sequence in a string**

#### **Java Program to find all the permutations of a string**

#### **Java Program to remove all the white spaces from a string**

#### **Java Program to replace lower-case characters with upper-case and vice-versa**

#### **Java Program to replace the spaces of a string with a specific character**

#### **Java Program to determine whether a given string is palindrome**

#### **Java Program to determine whether one string is a rotation of another**

#### **Java Program to find maximum and minimum occurring character in a string**

#### **Java Program to find Reverse each word of the string**

Here we are using index

**public** **class** ReverseWord {

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

String str = "123 456 789";

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

**int** j = 0, k = 0;

String rs = "";

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

//assign initial value of i to some variable

k=i;

// go upto space using i

**while**(i<arr.length&& arr[i]!=' ') {

i++;

}

// i have the index where we have white space

//assign that index to another variable i. e. k

j=i-1;

//now iterate from k to j and append to rs

**while**(k<=j) {

rs=rs+arr[j];

j--;

}

rs=rs+' ';

}

System.***out***.println(rs);

}

}

**Reverse each word of String using string array technique.**

1. convert into string array
2. go to each string element and make it reverse
3. append whitespace

**package** com.bs.strings;

**public** **class** Test09\_WordReverseUsingStringArray {

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

String string = "123 456 789";

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

**int** j = 0, k = 0;

String rs = "";

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

**char**[] arr = str[i].toCharArray();

j = arr.length - 1;

**while** (j >= 0) {

rs = rs + arr[j];

j--;

}

// add white space at end of above while loop

// to maintain white space. but extra white spac should

// not get added at last string

**if** (rs.length() < string.length())

rs = rs + ' ';

}

System.***out***.println(rs);

System.***out***.println(rs.length() + " " + string.length());

}

}

#### **Java program to find the duplicate characters in a string**

#### **Java program to find the duplicate words in a string**

#### **Java Program to find the frequency of characters**

#### **Java Program to find the largest and smallest word in a string**

**Difficult Program**

1. [Java Program to find the frequency of each element in the array](#_Java_Program_to)
2. [Java Program to left rotate the elements of an array](#_Java_Program_to_1)
3. Java Program to print the duplicate elements of an array

**Largest elements**

1. Jav[a Program to print the largest element in an array](#_Java_Program_to)
2. [Java Program to Find 3rd Largest Number in an array](#_Java_Program_to_2)
3. [Java Program to Find 2nd Largest Number in an array](#_Java_Program_to_1)
4. [Java Program to Find Largest Number in an array](#_Java_Program_to_3)

**Smallest elements**

1. [Java Program to print the smallest element in an array](#_Java_Program_to_4)
2. [Java to Program Find 2nd Smallest Number in an array](#_Java_to_Program)
3. [Java Program to Find Smallest Number in an array](#_Java_to_Program)

**Rotational elemnt**

1. Java Program to right rotate the elements of an array

**Arrays sorting**

1. [Java Program to sort the elements of an array in ascending order and descending order](#_Java_Program_to_5)
2. Java Program to Remove Duplicate Element in an array
3. [How to Sort an Array in Java](#_How_to_Sort)

## [Java Program to print the largest element in an array](https://www.javatpoint.com/java-program-to-print-the-largest-element-in-an-array)

**public** **class** Test06\_LargestElementInArray {

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

// Initialize array

**int**[] arr = **new** **int**[] { 25,12,10,02,2,5,65,98 };

**int** max = arr[0];

System.***out***.println("Duplicate elements in given array: ");

// Searches for duplicate element

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

**if** (arr[i] > max)

max = arr[i];

}

System.***out***.println("Max: " + max);

}

}

#### Java Program to Find 2nd Largest Number in an array

**package** com.programs.string;

**public** **class** Test06\_SecondLargestElementInArray {

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

// Initialize array

**int**[] a = **new** **int**[] { 1, 100,25, 65, 98, 52, 57 };

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

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

**if** (a[i] > a[j]) {

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

// sorted array

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

System.***out***.print(a[i]+" ");

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

System.***out***.println("2nd larget element : "+a[a.length-2]);

System.***out***.println("3rd larget element : "+a[a.length-3]); // line no. 1

}

}

#### Java Program to Find 3rd Largest Number in an array

**In this, we just need to print array 3rd last array element line no. 1 in above code**

#### Java Program to Find Largest Number in an array

**public** **class** Test06\_LargestElementInArray {

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

// Initialize array

**int**[] arr = **new** **int**[] { 25,12,10,02,2,5,65,98 };

**int** max = arr[0];

// Searches for duplicate element

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

**if** (arr[i] > max)

max = arr[i];

}

System.***out***.println("Max: " + max);

}

}

**OR**

**package** com.programs.string;

**public** **class** Test06\_LargestElementInArray {

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

// Initialize array

**int**[] a = **new** **int**[] { 25,12,10,02,2,5,65,98 };

//way 2

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

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

**if** (a[i] > a[j]) {

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

System.***out***.println("Lagest elemnt :"+a[a.length-1]);

}

}

**Smallest elements**

#### Java Program to print the smallest element in an array

**package** com.programs.string;

**public** **class** Test08\_SmallestElementInArray {

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

// Initialize array

**int**[] a = **new** **int**[] { 1, 100,25, 65, 98, 52, 57 };

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

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

**if** (a[i] < a[j]) {

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

// sorted array

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

System.***out***.print(a[i]+" ");

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

System.***out***.println("3rd smallest element : "+a[a.length-1]); // line no. 1

}

}

#### Java to Program Find 2nd and 3rd Smallest Number in an array

**public** **class** Test09\_ScondThridSmallestElementInArray {

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

// Initialize array

**int**[] a = **new** **int**[] { 1, 100,25, 65, 98, 52, 57 };

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

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

**if** (a[i] > a[j]) {

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

// sorted array

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

System.***out***.print(a[i]+" ");

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

System.***out***.println("2rd smallest element : "+a[1]); // line no. 1

System.***out***.println("3rd smallest element : "+a[2]); // line no. 1

}

}

**Arrays sorting**

#### Java Program to sort the elements of an array in ascending order and descending order

**public** **class** Test10\_SortArrayAscendingDescending {

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

// Initialize array

**int**[] a = **new** **int**[] { 1, 100,25, 65, 98, 52, 57 };

//ascending sort

System.***out***.println("Ascnding sorting");

*sortAscendingly*(a);

*showarray*(a);

//Desceding sorting

System.***out***.println("\nDesceding sorting");

*sortDescendinglly*(a);

*showarray*(a);

}

**private** **static** **void** sortDescendinglly(**int**[] a) {

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

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

**if** (a[i] < a[j]) {

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

**private** **static** **void** sortAscendingly(**int**[] a) {

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

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

**if** (a[i] > a[j]) {

**int** temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

**private** **static** **void** showarray(**int**[] a) {

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

System.***out***.print(a[i]+" ");

}

}

#### Java Program to Remove Duplicate Element in an array

public class **DuplicateCharFinder** {

    public void findIt(String str) {

        Map<Character, Integer> baseMap = new HashMap<Character, Integer>();

        char[] charArray = str.toCharArray();

        for (Character ch : charArray) {

            if (baseMap.containsKey(ch)) {

                baseMap.put(ch, baseMap.get(ch) + 1);

            } else {

                baseMap.put(ch, 1);

            }

        }

        Set<Character> keys = baseMap.keySet();

        for (Character ch : keys) {

            if (baseMap.get(ch) > 1) {

                System.out.println(ch + "  is " + baseMap.get(ch) + " times");

            }

        }

    }

    public static void main(String a[]) {

        DuplicateCharFinder dcf = new DuplicateCharFinder();

        dcf.findIt("India is my country");

    }

}

#### How to Sort an Array in Java

//invoking sort() method of the Arrays class  : Arrays.sort(array);

OR above code

public class **DuplicateCharFinder** {

    public void findIt(String str) {

        Map<Character, Integer> baseMap = new HashMap<Character, Integer>();

        char[] charArray = str.toCharArray();

        for (Character ch : charArray) {

            if (baseMap.containsKey(ch)) {

                baseMap.put(ch, baseMap.get(ch) + 1);

            } else {

                baseMap.put(ch, 1);

            }

        }

        Set<Character> keys = baseMap.keySet();

        for (Character ch : keys) {

            if (baseMap.get(ch) > 1) {

                System.out.println(ch + "  is " + baseMap.get(ch) + " times");

            }

        }

    }

    public static void main(String a[]) {

        DuplicateCharFinder dcf = new DuplicateCharFinder();

        dcf.findIt("India is my country");

    }

}