# Frequently Asked Java Programs for QA

Top 10 numbers Questions

1. **Swap two numbers Input:** a = 100, b= 200; **Output:** a = 200, b= 100;

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

**int** a = 100, b = 200;

System.***out***.println("After swapping, a = " + a + " and b = " + b);

// 1. Swapping using three Variables

**int** temp = a; a = b;

b = temp;

System.***out***.println("After swapping, a = " + a + " and b = " + b);

// 2. Using Two Variables a = a + b;

b = a - b;

a = a - b;

System.***out***.println("After swapping, a = " + a + " and b = " + b);

// 3. Swapping a and b using XOR a = a ^ b;

b = a ^ b; a = a ^ b;

System.***out***.println("After swapping, a = " + a + " and b = " + b);

}

# Armstrong number -

**Armstrong number is a number that is equal to the sum of cubes of its digits.**

**Input:** 153 , **Output:** Yes

153 is an Armstrong number. ==> (1\*1\*1) + (5\*5\*5) + (3\*3\*3) = 153

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

**int** sum = 0, res, temp;

**int** num = 153;// It is the number to check Armstrong temp = num;

**while** (num > 0) { res = num % 10; num = num / 10;

sum = sum + (res \* res \* res);

}

**if** (temp == sum)

System.***out***.println(temp + " is armstrong number");

# else

System.***out***.println(temp + " is Not armstrong number");

}

# Fibonacci Series –

**In Fibonacci series, next number is the sum of previous two numbers Input =** First 10 Numbers

**Output =** 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 etc.

The first two numbers of Fibonacci series are 0 and 1.

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

**int** num1 = 0, num2 = 1, num=10; **for** (**int** i = 0; i <= num; i++) { System.***out***.print(num1 + " ");

**int** num3 = num2 + num1;// Swap num1 = num2;

num2 = num3;

}

}

# Reverse a numbers and Number is Palindrome or Not.

**Input** = 12321

**Output** =12321

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

**int** num = 12321;

// 1. Reverse a Number Using the While Loop reversed number

**int** rev = 0;

**int** temp = num;

**int** rem; // remainder **while** (num > 0) { rem = num % 10;

rev = (rev \* 10) + rem; num = num / 10;

}

System.***out***.println("Reversed Number is " + rev);

// Verify number is palindrome or not

**if** (rev == temp) { System.***out***.println("palindrome number ");

} **else** {

System.***out***.println("not palindrome");

}

}

# Factorial Number

**Factorial Program in Java: Factorial of n is the product of all positive descending integers.**

# Input = 5!

**Output = 5! = 5\*4\*3\*2\*1 = 120**

**public static void** main(String[] args) { Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter number which you want for Factorial: ");

**int** num = sc.nextInt();

**int** fact = 1;

**for** (**int** i = 1; i <= num; i++) { fact = fact \* i;

}

System.***out***.println("Factorial of" + num + " is " + fact);

}

# OddEvenNumbers

**Input** = 11

**Output** = Given number is odd number

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

// 1. Using Brute Forcew Approach Scanner sc = **new** Scanner(System.in); System.out.println("Enter Number:-"); **int** num = sc.nextInt();

**if** (num % 2 == 0)// Brute Forcew Approach

{

System.out.println("Given is even number");

} **else** {

System.out.println("Given number is odd number");

}

# Prime Number

**Prime number is a number that is greater than 1 and divided by 1 or itself only. Input = 31, Output =** The number is prime.

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

**int** num = 31; **int** count = 0; **if** (num <= 1) {

System.***out***.println("The number is not prime");

# return;

}

**for** (**int** i = 2; i <= num / 2; i++) {

**if** (num % i == 0) count++;

}

**if** (count > 1) {

System.***out***.println("The number is not prime");

} **else** {

System.***out***.println("The number is prime"); }

# Largest number from 3 number/ given list

public static void main(String[] args) {

// TODO Auto-generated method stub

// 1. By using if else condition

int num1 = 7, num2 = 9, num3 = 10; if( num1 >= num2 && num1 >= num3)

System.out.println(num1 + " is the largest number."); else if (num2 >= num1 && num2 >= num3) System.out.println(num2 + " is the largest number."); else

System.out.println(num3 + " is the largest number.");

// 2. Using Collections.max() method and ArrayList ArrayList<Integer> x = new ArrayList<>(); x.add(12);

x.add(22);

x.add(54);

System.out.println(Collections.max(x)+ " is the largest number.");

}

# Sum of Digits

**Sum of all given numbers. Input** = 987

**Output** = 24

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

**int** n = 987; **int** sum = 0; **while** (n != 0) {

sum = sum + n % 10; n = n / 10;

}

System.***out***.println("Using While:- " + sum);

}

# Count digits in an integer number Input = 29845315, Output = 8

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

// **TODO** Auto-generated method stub

**long** num = 29845315;

**int** count = 0, num2 = 298453;

// 1. by using while loop

**while** (num != 0) {

num = num / 10; count++;

}

System.***out***.println("Number of digits : " + count);

// 2. Converting given number to string solution to count digits in an integer String result = Integer.*toString*(num2); // calculate the size of string System.***out***.println(+result.length());

}

Top 15 String Questions

1. Reverse a string

**Input** = mama

**Output** = mama

public static void main(String[] args) { String str = "mama";

# String s2 = "";

**// 1. by using the charAt() method**

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

s2 = s2 + str.charAt(i);// extracts each character and store in string

}

System.out.println("Reversed word: " + s2);

# // below is code to check weather given string is Palindrome or not

if (str.equalsIgnoreCase(**s2**)) { System.*out*.println("String is Palindrome");

} else {

System.*out*.println("String is not Palindrome");

}

}

# // 2. Using built in reverse() method of the StringBuilder class:

String input = "Welcome To Jave Learning"; StringBuilder input1 = new StringBuilder();

input1.append(input); // append a string into StringBuilder input1 input1.reverse();

System.*out*.println(input1);

# // 3. Using StringBuffer:

String strText = "Java Learning";

// conversion from String object to StringBuffer StringBuffer sbr = new StringBuffer(strText); sbr.reverse();

System.*out*.println(sbr);

1. Remove space form given string

**Input String** = “hello java Learning ”

**Output String** = “hellojavaLearning”

**public static void** main(String[] args) { System.***out***.println("Enter String "); Scanner sc = **new** Scanner(System.***in***); String input = sc.nextLine();

System.***out***.println("Original String- " + input); input = input.replaceAll("\\s", ""); System.***out***.println("Final String- " + input);

}

1. Finding Common Elements in Arrays

# Input =

array1 = { 4, 2, 3, 1, 6 }; array2 = { 6, 7, 8, 4 };

**Output** = 6,4

// By using the for loop

Integer[] array1 = { 4, 2, 3, 1, 6 };

Integer[] array2 = { 6, 7, 8, 4 };

List<Integer> commonElements = **new** ArrayList<>();

**for** (**int** i = 0; i < array1.length; i++) { **for** (**int** j = 0; j < array2.length; j++) { **if** (array1[i] == array2[j]) { commonElements.add(array1[i]);

} }}

System.***out***.println("Common Elements for given two array List is:" + commonElements);

// by using ArrayList with retainAll method

ArrayList<Integer> list1 = **new** ArrayList<>(Arrays.*asList*(array1)); ArrayList<Integer> list2 = **new** ArrayList<>(Arrays.*asList*(array2)); list1.retainAll(list2);

System.***out***.println("Common Elements:" + list1);

// By using Java Stream

String[] array3 = { "Java", "JavaScript", "C", "C++" };

String[] array4 = { "Python", "C#", "Java", "C++" }; ArrayList<String> list3 = **new** ArrayList<>(Arrays.*asList*(array3)); ArrayList<String> list4 = **new** ArrayList<>(Arrays.*asList*(array4)); List<String> commonElements1 = list3.stream().filter(list4::contains).collect(Collectors.*toList*()); System.***out***.println(commonElements1);

} }

1. Find first and last element of ArrayList in java

**Input =** array1 = { 4, 2, 3, 1, 6 };

# Output = First is:4, Last is: 6

**ArrayList<Integer> list = new ArrayList<Integer>(5);**

# // find first element

int first = list.get(0);//First Element

# // find last element

int last = list.get(list.size() - 1);//last Element

1. Second Largest and Second Smallest Numbers:

// Code to find second largest and second smallest numbers in an array

**int**[] arrayList = { 4, 2, 3, 1,0, 6,12,15,20 };

**int** num=arrayList.length; Arrays.*sort*(arrayList);

System.***out***.println("Second Largest element is "+arrayList[num-2]); //Display Second Smallest

System.***out***.println("Second Smallest element is "+arrayList[1]);

1. How to sort an Array without using inbuilt method?

**Input** = array[] = { 10, 5, 20, 63, 12, 57, 88, 60 };

**Output** = 5 10 12 20 57 60 63 88

int temp, size;

int array[] = { 10, 5, 20, 63, 12, 57, 88, 60 };

size = array.length;

for (int i = 0; i < size; i++) { for (int j = i + 1; j < size; j++) { if (array[i] > array[j]) {

temp = array[i]; array[i] = array[j]; array[j] = temp;

}}}

for (int i = 0; i < array.length; i++) { System.out.println("Array sorted: " + array[i]);

}

**// Print 3rd Largest number from an Array** System.out.println("Third largest number is:: " + array[size - 3]); System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

# // sort array using the Arrays.sort method

Arrays.sort(array);

System.out.println("sorted array- " + Arrays.toString(array)); int thirdMaxNum=array[size-3];

System.out.println("Third highest array- " +thirdMaxNum );

1. Counting number of occurrences of given word in a string using Java?

**String** = "Java is a programming language. Java is widely used in software Testing";

**Input** = ”Java”, **Output** = 2

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

String string = "Java is a programming language. Java is widely used in software Testing";

String[] words = string.toLowerCase().split(" "); String word = "java";

**int** occurrences = 0;

**for** (**int** i = 0; i < words.length; i++) **if** (words[i].equals(word)) occurrences++; System.***out***.println(occurrences);

}

1. Find each word occurrence from given string in string java

**Input** = "Alice is girl and Bob is boy";

**Output** = {Bob=1, Alice=1, and=1, is=2, girl=1, boy=1}

**public static void** main(String[] args) { String str = "Alice is girl and Bob is boy";

Map<String, Integer> hashMap = **new** HashMap<>(); String[] words = str.split(" ");

**for** (String word : words) {

**if** (hashMap.containsKey(word)) hashMap.put(word, hashMap.get(word) + 1); **else**

hashMap.put(word, 1);

}

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

1. Reverse the entire sentence **Input** = "India is country My" **Output** = "My country is India"

**public static void** main(String[] args) { String str[] = "India is country My".split(" "); String ans = "";

**for** (**int** i = str.length - 1; i >= 0; i--) { ans = ans + str[i] + " ";

}

System.***out***.println(ans.substring(0, ans.length() - 1));

}

1. count the occurrences of each character?

**Input** = "This is an example";

**Output** = p = 1, a = 2, s = 2, T = 1, e = 2, h = 1, x = 1, i = 2, l = 1, m = 1, n = 1

**public static void** main(String[] args) { String str = "This is an example";

HashMap<Character, Integer> count = **new** HashMap<Character, Integer>();

// convert string to character array

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

// traverse every character and count the Occurrences

**for** (**char** c : arr) {

// filter out white spaces

**if** (c != ' ') {

**if** (count.containsKey(c)) {

// if character already traversed, increment it count.put(c, count.get(c) + 1);

} **else** {

// if character not traversed, add it to hashmap count.put(c, 1);

}

}

}

// traverse the map and print the number of occurences of a character

**for** (Map.Entry entry : count.entrySet()) {

System.***out***.print( entry.getKey() + " = " + entry.getValue()+", ");

}

}

1. Removing Duplicates from an Array

// using for loop

String[] strArray = {"abc", "def", "abc", "mno", "xyz", "pqr", "xyz", "pqr"};

//1. Using Brute Force Method

**for** (**int** i = 0; i < strArray.length-1; i++)

{

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

{

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

{

System.***out***.println("Brute Force Method : Duplicate Element is : "+strArray[j]);

}}}

// using Hashset

HashSet<String> hs = **new** HashSet<String>();

**for** (String arrayElement : strArray)

{

**if**(!hs.add(arrayElement))

{System.***out***.println("HashSet :Duplicate Element is : "+arrayElement);

}}

1. Reverse each word in a sentence **Input** = "reverse each word in a string"; **Output** = "esrever hcae drow ni a gnirts"

**public static void** main(String[] args) { String str = "reverse each word in a string"; String words[] = str.split("\\s");

String reverseWord = "";

**for** (String w : words) {

StringBuilder sb = **new** StringBuilder(w); sb.reverse();

reverseWord = reverseWord + sb.toString() + " ";

}

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

}

1. String Anagrams: Determine if two strings are anagrams of each other

# Input =

String str1 = "Army"; String str2 = "Mary";

**Output** = **army** and **mary** are anagram.

**public static void** main(String[] args) { String str1 = "Army";

String str2 = "Mary";

str1 = str1.toLowerCase(); str2 = str2.toLowerCase();

// check if length is same

**if** (str1.length() == str2.length()) {

// convert strings to char array

**char**[] charArray1 = str1.toCharArray();

**char**[] charArray2 = str2.toCharArray();

// sort the char array Arrays.*sort*(charArray1); Arrays.*sort*(charArray2);

// if sorted char arrays are same, then the string is anagram

**boolean** result = Arrays.*equals*(charArray1, charArray2);

**if** (result) {

System.***out***.println(str1 + " and " + str2 + " are anagram.");

} **else** {

System.***out***.println(str1 + " and " + str2 + " are not anagram.");

}

} **else** {

System.***out***.println(str1 + " and " + str2 + " are not anagram.");

}

1. How to print duplicate characters from the string?

**Input** = "apple is fruit";

**Output** = p i

**public static void** main(String[] args) { String str = "apple is fruit";

**char**[] carray = str.toCharArray(); System.***out***.println("The string is:" + str);

System.***out***.print("Duplicate Characters in above string are: ");

**for** (**int** i = 0; i < str.length(); i++) { **for** (**int** j = i + 1; j < str.length(); j++) { **if** (carray[i] == carray[j]) { System.***out***.print(carray[j] + ""); **break**;

}

}

}

}

1. Find and print the largest element in an array.

// Initialize array

**int**[] arr = **new int**[] { 25, 11, 7, 75, 56 };

// Initialize max with first element of array.

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

// Loop through the array

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

// Compare elements of array with max

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

}

System.***out***.println("Largest element present in given array: " + max);

1. Java program to split an alphanumeric digit without using split method

**Input** = "Welcome234To567Java89Programming0@#!!";

**Output** = WelcomeToJavaProgramming 234567890

@#!!

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

String str = "Welcome234To567Java89Programming0@#!!";

StringBuffer alpha = **new** StringBuffer(), num = **new** StringBuffer(), special = **new**

StringBuffer();

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

**if** (Character.*isDigit*(str.charAt(i))) num.append(str.charAt(i));

**else if** (Character.*isAlphabetic*(str.charAt(i))) alpha.append(str.charAt(i));

# else

special.append(str.charAt(i));

}

System.***out***.println(alpha); System.***out***.println(num); System.***out***.println(special);

}