Write a program to Reverse a String

**package** interviewProgms;

// To Reverse a String

// Difference between String and String buffer

// Do we have reverse function in String? No

**public** **class** reverseString {

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

String s = "Welcome";

//1. using for loop

**int** len=s.length();

String rev="";

**for**(**int** i=len-1; i>=0; i--) {

rev=rev+s.charAt(i);

}

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

//2. using StringBuffer class:

StringBuffer sf=**new** StringBuffer(s);

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

}

}

Write a program to Remove Junk/Special Characters in a String

**package** interviewProgms;

//To Remove Junk/Special Characters in a String

//Remove special chars/Chinese/Japanese chars from a String

**public** **class** RemoveJunk\_SpecialCharacters {

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

String s = "漢字汉字 String 0123456789&&&@@#$#$#java";

//Regular expression: [^a-zA-z0-9]

s = s.replaceAll("[^a-zA-Z0-9]", "");

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

}

}

Write a program to find common elements between two arrays.

class comElementArray{

public static void main(String[] args){

int[] arr1 ={12,23,34,45,90};

int[] arr2 ={34,546,78,90,12};

for(int i=0;i<=arr1.length-1;i++){

for(int j=0;j<=arr2.length-1;j++){

if(arr1[i]==arr2[j])

{

System.out.println(arr1[i]);

}

}

}

}}

Write a program to find top two maximum numbers in a array.

class maxTwo{

public static void main(String[] args){

int[] arr = {12,23,1,45,67,22};

int max=0;

int max2=0;

for (int i:arr){

if(i>max){

max2=max;

max=i;

}

}

System.out.println(max);

System.out.println(max2);

}

}

Write a program to find smallest and largest number in a array.

**package** interviewProgms;

**import** java.util.Arrays;

// To find Smallest and Largest Number in a Array

**public** **class** SmllestAndLargestNumber {

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

**int** a[]= {10, 48, -101, 25, 6845};

**int** largest = a[0];

**int** smallest = a[0];

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

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

largest = a[i];

}

**else** **if**(a[i]<smallest) {

smallest = a[i];

}

}

System.***out***.println("\n given Array is:" + Arrays.*toString*(a));

System.***out***.println("Largest number is:" + largest);

System.***out***.println("Smallest number is:" + smallest);

}

}

Write a program to find Duplicates Elements in Java Array.

**package** interviewProgms;

**import** java.util.HashSet;

**import** java.util.Set;

**public** **class** DuplicatElementArray {

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

String names[]= {"Java", "Javascript", "Ruby", "C", "Python", "Java", "Ruby"};

//1. Compare each element:O(nxn) --- worst solution

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

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

**if**(names[i].equals(names[j])) {

System.***out***.println("duplicate element is: " + names[i]);

}

}

}

//2. Using HashSet: Java collection - it stores unique value: O(n)

Set<String> store = **new** HashSet<String>();

**for**(String name:names) {

**if**(store.add(name)==**false**) {

System.***out***.println("duplicate element is:" + name);

}

}

}

}

to convert decimal to binary

System.out.println("Binary representation of 124: ");

System.out.println(Integer.toBinaryString(124));

### Without using predefined method

class DecimalBinaryExample{

public void convertBinary(int num){

int binary[] = new int[40];

int index = 0;

while(num > 0){

binary[index++] = num%2;

num = num/2;

}

for(int i = index-1;i >= 0;i--){

System.out.print(binary[i]);

}

}

public static void main(String a[]){

DecimalBinaryExample obj = new DecimalBinaryExample();

System.out.println("Binary representation of 124: ");

obj.convertBinary(124);

System.out.println("\nBinary representation of 45: ");

obj.convertBinary(45);

System.out.println("\nBinary representation of 999: ");

obj.convertBinary(999);

}

}

decimal to binary using stack

import java.util.\*;

class DecimalBinaryStack

{

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

// Create Stack object

Stack<Integer> stack = new Stack<Integer>();

// User input

System.out.println("Enter decimal number: ");

int num = in.nextInt();

while (num != 0)

{

int d = num % 2;

stack.push(d);

num /= 2;

}

System.out.print("\nBinary representation is:");

while (!(stack.isEmpty() ))

{

System.out.print(stack.pop());

}

System.out.println();

}

}

Swap without using third variable

Int a;

Int b;

a=a+b;

b=a-b;

a=a-b;

#### Program to check number is even or odd by using division “%” "/" operator

1. package instanceofjava;
2. import java.util.Scanner;
4. public class EvenorOdd {
6. public static void main(String []args )    {
8. int number;
9. Scanner in= new Scanner(System.in);
11. System.out.println("Enter a number to check even or odd");
12. number=in.nextInt();
14. if((number & 1)==0){
15. System.out.println(+number+" is Even number");
16. }else{
17. System.out.println(+number+" is Odd Number");
18. }
20. }
21. }

program to Reverse a Number

**while**(n != 0)

{

reverse = reverse \* 10;

reverse = reverse + n%10;

n = n/10;

}

Palindrome

class Palindrome

{

public static void main(String args[])

{

String original, reverse = ""; // Objects of String class

Scanner in = new Scanner(System.in);

System.out.println("Enter a string to check if it is a palindrome");

original = in.nextLine();

int length = original.length();

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

reverse = reverse + original.charAt(i);

if (original.equals(reverse))

System.out.println("Entered string is a palindrome.");

else

System.out.println("Entered string isn't a palindrome.");

}

}

## Fibonacci Series without using recursion

1. class FibonacciExample1{
2. public static void main(String args[])
3. {
4. int n1=0,n2=1,n3,i,count=10;
5. System.out.print(n1+" "+n2);//printing 0 and 1
7. for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed
8. {
9. n3=n1+n2;
10. System.out.print(" "+n3);
11. n1=n2;
12. n2=n3;
13. }
15. }}

Fibonacci Series using recursion in

class FibonacciExample2{

1. static int n1=0,n2=1,n3=0;
2. static void printFibonacci(int count){
3. if(count>0){
4. n3 = n1 + n2;
5. n1 = n2;
6. n2 = n3;
7. System.out.print(" "+n3);
8. printFibonacci(count-1);
9. }
10. }
11. public static void main(String args[]){
12. int count=10;
13. System.out.print(n1+" "+n2);//printing 0 and 1
14. printFibonacci(count-2);//n-2 because 2 numbers are already printed
15. }
16. }

Armstrong Number: the sum of cubes of its digits equals to number

1. class ArmstrongExample{
2. public static void main(String[] args)  {
3. int c=0,a,temp;
4. int n=153;//It is the number to check armstrong
5. temp=n;
6. while(n>0)
7. {
8. a=n%10;
9. n=n/10;
10. c=c+(a\*a\*a);
11. }
12. if(temp==c)
13. System.out.println("armstrong number");
14. else
15. System.out.println("Not armstrong number");
16. }
17. }

# Find out duplicate number between 1 to N numbers.

package com.java2novice.algos;

import java.util.ArrayList;

import java.util.List;

public class DuplicateNumber {

    public int findDuplicateNumber(List<Integer> numbers){

        int highestNumber = numbers.size() - 1;

        int total = getSum(numbers);

        int duplicate = total - (highestNumber\*(highestNumber+1)/2);

        return duplicate;

    }

    public int getSum(List<Integer> numbers){

        int sum = 0;

        for(int num:numbers){

            sum += num;

        }

        return sum;

    }

    public static void main(String a[]){

        List<Integer> numbers = new ArrayList<Integer>();

        for(int i=1;i<30;i++){

            numbers.add(i);

        }

        //add duplicate number into the list

        numbers.add(22);

        DuplicateNumber dn = new DuplicateNumber();

        System.out.println("Duplicate Number: "+dn.findDuplicateNumber(numbers));

    }

}

# Find out middle index where sum of both ends are equal

public class FindMiddleIndex {

    public static int findMiddleIndex(int[] numbers) throws Exception {

        int endIndex = numbers.length - 1;

        int startIndex = 0;

        int sumLeft = 0;

        int sumRight = 0;

        while (true) {

            if (sumLeft > sumRight) {

                sumRight += numbers[endIndex--];

            } else {

                sumLeft += numbers[startIndex++];

            }

            if (startIndex > endIndex) {

                if (sumLeft == sumRight) {

                    break;

                } else {

                    throw new Exception(

                            "Please pass proper array to match the requirement");

                }

            }

        }

        return endIndex;

    }

    public static void main(String a[]) {

        int[] num = { 2, 4, 4, 5, 4, 1 };

        try {

            System.out.println("Starting from index 0, adding numbers till index "

                            + findMiddleIndex(num) + " and");

            System.out.println("adding rest of the numbers can be equal");

        } catch (Exception ex) {

            System.out.println(ex.getMessage());

        }

    }

}

Write a singleton class

public class MySingleton {

    private static MySingleton myObj;

    static{

        myObj = new MySingleton();

    }

    private MySingleton(){

    }

    public static MySingleton getInstance(){

        return myObj;

    }

    public void testMe(){

        System.out.println("Hey.... it is working!!!");

    }

    public static void main(String a[]){

        MySingleton ms = getInstance();

        ms.testMe();

    }

}

Write a program to reverse a string using recursive algorithm

public class StringRecursiveReversal {

    String reverse = "";

    public String reverseString(String str){

        if(str.length() == 1){

            return str;

        } else {

            reverse += str.charAt(str.length()-1)

                    +reverseString(str.substring(0,str.length()-1));

            return reverse;

        }

    }

    public static void main(String a[]){

        StringRecursiveReversal srr = new StringRecursiveReversal();

        System.out.println("Result: "+srr.reverseString("Java2novice"));

    }

}

Write a program to find perfect number or not.

class perfectNumber{

public static void main(String[] args){

java.util.Scanner scn = new java.util.Scanner(System.in);

System.out.println("Enter a number");

int num=scn.nextInt();

if(num==isPerfect(num)){

System.out.println(num+" is perfect number");

}

else{

System.out.println(num+" is not a perfect number");

}

}

static int isPerfect(int num){

int sum=0;

for(int i=1;i<num-1;i++){

if(num%i==0){

sum+=i;

}

}

return sum;

}

}

Wrie a program to find out duplicate characters in a string.

class duplicateChar{

public static void main(String[] args){

java.util.Scanner scn = new java.util.Scanner(System.in);

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

String str = scn.next();

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

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

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

System.out.println(str.charAt(i));

}

}

}

}

}

Write a program to find sum of each digit in the given number using recursion.

class findRecurtion

{

public int sum=0;

java.util.Scanner scn = new java.util.Scanner(System.in);

public int num = scn.nextInt();

public static void main(String[] args)

{

System.out.println("Enter a number");

findRecurtion obj =new findRecurtion();

obj.sum();

System.out.println("Sum of number is: "+obj.sum);

}

public void sum()

{ if(num>0){

sum +=num%10;

num =num/10;

sum();

}

}

}

String Manipulation:

**package** interviewProgms;

**public** **class** String\_Manipulation {

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

String str = "Welcome to Goa Singham";

String str1 = "welcome to Goa Singham";

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

System.***out***.println(str.charAt(5));

System.***out***.println(str.indexOf('o')); // 1st occurrence of o

System.***out***.println(str.indexOf('o',5)); // 2nd occurrence of o (Hard coded)

System.***out***.println(str.indexOf('o',str.indexOf('o')+1));// 2nd occurrence of o (generic)

System.***out***.println(str.indexOf("Goa"));

System.***out***.println(str.indexOf("Mumbai"));// -1

//String comparison

System.***out***.println(str.equals(str1));

System.***out***.println(str.equalsIgnoreCase(str1));

//subString

System.***out***.println(str.substring(0, 8));

//trim

String s = " Hello world ";

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

System.***out***.println(s.replace(" ",""));

String date = "01-01-2019";

System.***out***.println(date.replace("-", "/")); //01/01/2019

//split

String test = "Neil\_Nitin\_Mukesh";

String[] testVal = test.split("\_");

**for**(String s1:testVal) {

System.***out***.println(s1);

}

//concat:

String s2="Deepika";

System.***out***.println(s2.concat("Ranveer"));

String x = "Hello";

String y = "world";

**int** a = 100;

**int** b = 200;

System.***out***.println(x+y);

System.***out***.println(a+b);

System.***out***.println(x+y+a+b);

System.***out***.println(a+b+x+y);

System.***out***.println(x+y+(a+b));

}

}

**Output:**

22

m

4

9

9

11

-1

false

true

Welcome

Hello world

Helloworld

01/01/2019

Neil

Nitin

Mukesh

DeepikaRanveer

Helloworld

300

Helloworld100200

300Helloworld

Helloworld300

Write a program to find Missing Number In Array

**package** interviewProgms;

// How to find Missing Number In Array

**public** **class** MissingNumber\_Array {

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

**int** a[] = {1,2,3,4,5,7,8,9,10};

**int** sum = 0;

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

sum = sum + a[i];//49

}

**int** sum1 = 0;

**for**(**int** j=1; j<=10; j++) {

sum1 = sum1 + j;//55

}

System.***out***.println("missing number is::" + (sum1-sum));

}

}

Write a program to Swap two Strings without using temp/third variable

**package** interviewProgms;

**public** **class** SwapStringswithoutThirdvariable {

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

// WAP to Swap two Strings without using temp/third variable

String a = "Hello";

String b = "World";

a = a+b;

b=a.substring(0, a.length()-b.length());

a=a.substring(b.length());

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

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

}

}