**Java program to reverse a number using for, while and recursion**

There are three ways to reverse a number in Java.

1) [**Using while loop**](http://beginnersbook.com/2014/01/java-program-to-reverse-a-number/#using-while)

2) [**Using for loop**](http://beginnersbook.com/2014/01/java-program-to-reverse-a-number/#using-for)

3) [**Using recursion**](http://beginnersbook.com/2014/01/java-program-to-reverse-a-number/#using-recursion)

4) [**Reverse the number without user interaction**](http://beginnersbook.com/2014/01/java-program-to-reverse-a-number/#demo)

**Program 1: Reverse a number using while Loop**

The program will prompt user to input the number and then it will reverse the same number using while loop.

import java.util.Scanner;  
class ReverseNumberWhile  
{  
 public static void main(String args[])  
 {  
 int num=0;  
 int reversenum =0;  
 System.out.println("Input your number and press enter: ");  
 //This statement will capture the user input  
 Scanner in = new Scanner(System.in);  
 //Captured input would be stored in number num  
 num = in.nextInt();  
 //While Loop: Logic to find out the reverse number  
 while( num != 0 )  
 {  
 reversenum = reversenum \* 10;  
 reversenum = reversenum + num%10;  
 num = num/10;  
 }  
  
 System.out.println("Reverse of input number is: "+reversenum);  
 }  
}

Output:

Input your number and press enter:   
145689  
Reverse of input number is: 986541

-------------------------------------------------------------------------------------

**Program 2: Reverse a number using for Loop**

import java.util.Scanner;  
class ForLoopReverseDemo  
{  
 public static void main(String args[])  
 {  
 int num=0;  
 int reversenum =0;  
 System.out.println("Input your number and press enter: ");  
 //This statement will capture the user input  
 Scanner in = new Scanner(System.in);  
 //Captured input would be stored in number num  
 num = in.nextInt();  
 /\* for loop: No initialization part as num is already  
 \* initialized and no increment/decrement part as logic  
 \* num = num/10 already decrements the value of num  
 \*/  
 for( ;num != 0; )  
 {  
 reversenum = reversenum \* 10;  
 reversenum = reversenum + num%10;  
 num = num/10;  
 }  
  
 System.out.println("Reverse of specified number is: "+reversenum);  
 }  
}

Output:

Input your number and press enter:   
56789111  
Reverse of specified number is: 11198765

-------------------------------------------------------------------------------------

**Program 3: Reverse a number using recursion**

import java.util.Scanner;  
class RecursionReverseDemo  
{  
 //A method for reverse  
 public static void reverseMethod(int number) {  
 if (number < 10) {  
 System.out.println(number);  
 return;  
 }  
 else {  
 System.out.print(number % 10);  
 //Method is calling itself: recursion  
 reverseMethod(number/10);  
 }  
 }  
 public static void main(String args[])  
 {  
 int num=0;  
 System.out.println("Input your number and press enter: ");  
 Scanner in = new Scanner(System.in);  
 num = in.nextInt();  
 System.out.print("Reverse of the input number is:");  
 reverseMethod(num);  
 System.out.println();  
 }  
}

Output:

Input your number and press enter:   
5678901  
Reverse of the input number is:1098765

Example: Reverse an already initialized number

In all the above programs we are prompting user for the input number, however if do not want the user interaction part and want to reverse an initialized number then this is how you can do it.

class ReverseNumberDemo  
{  
 public static void main(String args[])  
 {  
 int num=123456789;  
 int reversenum =0;  
 while( num != 0 )  
 {  
 reversenum = reversenum \* 10;  
 reversenum = reversenum + num%10;  
 num = num/10;  
 }  
  
 System.out.println("Reverse of specified number is: "+reversenum);  
 }  
}

Output:

Reverse of specified number is: 987654321

-------------------------------------------------------------------------------------

J***ava Program to calculate area and circumference of circle***

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JANUARY 18, 2014

In this tutorial we will see how to calculate area and circumference of circle in Java. There are two ways to do this:

1) With user interaction: Program will prompt user to enter the radius of the circle

2) Without user interaction: The radius value would be specified in the program itself.

Program 1:

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: Program to calculate area and circumference of circle  
 \* with user interaction. User will be prompt to enter the radius and   
 \* the result will be calculated based on the provided radius value.  
 \*/  
import java.util.Scanner;  
class CircleDemo  
{  
 static Scanner sc = new Scanner(System.in);  
 public static void main(String args[])  
 {  
 System.out.print("Enter the radius: ");  
 /\*We are storing the entered radius in double  
 \* because a user can enter radius in decimals  
 \*/  
 double radius = sc.nextDouble();  
 //Area = PI\*radius\*radius  
 double area = Math.PI \* (radius \* radius);  
 System.out.println("The area of circle is: " + area);  
 //Circumference = 2\*PI\*radius  
 double circumference= Math.PI \* 2\*radius;  
 System.out.println( "The circumference of the circle is:"+circumference) ;  
 }  
}

Output:

Enter the radius: 1  
The area of circle is: 3.141592653589793  
The circumference of the circle is:6.283185307179586

Program 2:

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: Program to calculate area and circumference of circle  
 \* without user interaction. You need to specify the radius value in   
 \* program itself.  
 \*/  
class CircleDemo2  
{  
 public static void main(String args[])  
 {  
 int radius = 3;  
 double area = Math.PI \* (radius \* radius);  
 System.out.println("The area of circle is: " + area);  
 double circumference= Math.PI \* 2\*radius;  
 System.out.println( "The circumference of the circle is:"+circumference) ;  
 }  
}

Output:

The area of circle is: 28.274333882308138  
The circumference of the circle is:18.84955592153876

-------------------------------------------------------------------------------------

Java program to calculate area of Triangle

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JANUARY 18, 2014

Here we will see how to calculate area of triangle. We will see two following programs to do this:

**1) Program 1: Prompt user for base-width and height of triangle.**

**2) Program 2: No user interaction: Width and height are specified in the program itself.**

Program 1:

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: Program to Calculate area of Triangle in Java  
 \* with user interaction. Program will prompt user to enter the   
 \* base width and height of the triangle.  
 \*/  
import java.util.Scanner;  
class AreaTriangleDemo {  
 public static void main(String args[]) {   
 Scanner scanner = new Scanner(System.in);  
  
 System.out.println("Enter the width of the Triangle:");  
 double base = scanner.nextDouble();  
  
 System.out.println("Enter the height of the Triangle:");  
 double height = scanner.nextDouble();  
  
 //Area = (width\*height)/2  
 double area = (base\* height)/2;  
 System.out.println("Area of Triangle is: " + area);   
 }  
}

Output:

Enter the width of the Triangle:  
2  
Enter the height of the Triangle:  
2  
Area of Triangle is: 2.0

Program 2:

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: Program to Calculate area of Triangle  
 \* with no user interaction.  
 \*/  
class AreaTriangleDemo2 {  
 public static void main(String args[]) {   
 double base = 20.0;  
 double height = 110.5;  
 double area = (base\* height)/2;  
 System.out.println("Area of Triangle is: " + area);   
 }  
}

Output:

Area of Triangle is: 1105.0

-------------------------------------------------------------------------------------

**Java program to sum the elements of an array**

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JANUARY 18, 2014

In this tutorial we will see how to sum up all the elements of an array.

Program 1: No user interaction

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: Get sum of array elements  
 \*/  
class SumOfArray{  
 public static void main(String args[]){  
 int[] array = {10, 20, 30, 40, 50, 10};  
 int sum = 0;  
 //Advanced for loop  
 for( int num : array) {  
 sum = sum+num;  
 }  
 System.out.println("Sum of array elements is:"+sum);  
 }  
}

Output:

Sum of array elements is:160

Program 2: User enters the array’s elements

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: User would enter the 10 elements  
 \* and the program will store them into an array and   
 \* will display the sum of them.  
 \*/  
import java.util.Scanner;  
class SumDemo{  
 public static void main(String args[]){  
 Scanner scanner = new Scanner(System.in);  
 int[] array = new int[10];  
 int sum = 0;  
 System.out.println("Enter the elements:");  
 for (int i=0; i<10; i++)  
 {  
 array[i] = scanner.nextInt();  
 }  
 for( int num : array) {  
 sum = sum+num;  
 }  
 System.out.println("Sum of array elements is:"+sum);  
 }  
}

Output:

Enter the elements:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Sum of array elements is:55

-------------------------------------------------------------------------------------

**Java program to check prime number**

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JANUARY 18, 2014

This program will prompt user to enter a number and then it will check and display whether the input number is prime or not.

import java.util.Scanner;  
class PrimeCheck  
{  
 public static void main(String args[])  
 {   
 int temp;  
 boolean isPrime=true;  
 Scanner scan= new Scanner(System.in);  
 System.out.println("Enter a number for check:");  
 //capture the input in an integer  
 int num=scan.nextInt();  
 for(int i=2;i<=num/2;i++)  
 {  
 temp=num%i;  
 if(temp==0)  
 {  
 isPrime=false;  
 break;  
 }  
 }  
 //If isPrime is true then the number is prime else not  
 if(isPrime)  
 System.out.println(num + " is Prime Number");  
 else  
 System.out.println(num + " is not Prime Number");  
 }  
}

Output:

Enter a number for check:  
19  
19 is Prime Number

Output 2:

Enter a number for check:  
6  
6 is not Prime Number

-------------------------------------------------------------------------------------

**Java Program to check Even or Odd number**

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: FEBRUARY 11, 2014

import java.util.Scanner;  
  
class CheckEvenOdd  
{  
 public static void main(String args[])  
 {  
 int num;  
 System.out.println("Enter an Integer number:");  
  
 //The input provided by user is stored in num  
 Scanner input = new Scanner(System.in);  
 num = input.nextInt();  
  
 /\* If number is divisible by 2 then it's an even number  
 \* else odd number\*/  
 if ( num % 2 == 0 )  
 System.out.println("Entered number is even");  
 else  
 System.out.println("Entered number is odd");  
 }  
}

Output 1:

Enter an Integer number:  
78  
Entered number is even

Output 2:

Enter an Integer number:  
77  
Entered number is odd

-------------------------------------------------------------------------------------

**Java program for linear search – Example**

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: APRIL 9, 2014

Example Program:

This program uses [**linear search algorithm**](http://en.wikipedia.org/wiki/Linear_search) to find out a number among all other numbers entered by user.

/\* Program: Linear Search Example  
 \* Written by: Chaitanya from beginnersbook.com  
 \* Input: Number of elements, element's values, value to be searched  
 \* Output:Position of the number input by user among other numbers\*/  
import java.util.Scanner;  
class LinearSearchExample  
{  
 public static void main(String args[])  
 {  
 int counter, num, item, array[];  
 //To capture user input  
 Scanner input = new Scanner(System.in);  
 System.out.println("Enter number of elements:");  
 num = input.nextInt();   
 //Creating array to store the all the numbers  
 array = new int[num];   
 System.out.println("Enter " + num + " integers");  
 //Loop to store each numbers in array  
 for (counter = 0; counter < num; counter++)  
 array[counter] = input.nextInt();  
  
 System.out.println("Enter the search value:");  
 item = input.nextInt();  
  
 for (counter = 0; counter < num; counter++)  
 {  
 if (array[counter] == item)   
 {  
 System.out.println(item+" is present at location "+(counter+1));  
 /\*Item is found so to stop the search and to come out of the   
 \* loop use break statement.\*/  
 break;  
 }  
 }  
 if (counter == num)  
 System.out.println(item + " doesn't exist in array.");  
 }  
}

Output 1:

Enter number of elements:  
6  
Enter 6 integers  
22  
33  
45  
1  
3  
99  
Enter the search value:  
45  
45 is present at location 3

Output 2:

Enter number of elements:  
4  
Enter 4 integers  
11  
22  
4  
5  
Enter the search value:  
99  
99 doesn't exist in array.

-------------------------------------------------------------------------------------

Java program to perform binary search – Example

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: APRIL 21, 2014

Example Program to perform binary search on a list of integer numbers

This program uses [**binary search algorithm**](http://en.wikipedia.org/wiki/Binary_search_algorithm) to search an element in given list of elements.

/\* Program: Binary Search Example  
 \* Written by: Chaitanya from beginnersbook.com  
 \* Input: Number of elements, element's values, value to be searched  
 \* Output:Position of the number input by user among other numbers\*/  
import java.util.Scanner;  
class BinarySearchExample  
{  
 public static void main(String args[])  
 {  
 int counter, num, item, array[], first, last, middle;  
 //To capture user input  
 Scanner input = new Scanner(System.in);  
 System.out.println("Enter number of elements:");  
 num = input.nextInt();   
  
 //Creating array to store the all the numbers  
 array = new int[num];  
  
 System.out.println("Enter " + num + " integers");  
 //Loop to store each numbers in array  
 for (counter = 0; counter < num; counter++)  
 array[counter] = input.nextInt();  
  
 System.out.println("Enter the search value:");  
 item = input.nextInt();  
 first = 0;  
 last = num - 1;  
 middle = (first + last)/2;  
  
 while( first <= last )  
 {  
 if ( array[middle] < item )  
 first = middle + 1;  
 else if ( array[middle] == item )  
 {  
 System.out.println(item + " found at location " + (middle + 1) + ".");  
 break;  
 }  
 else  
 {  
 last = middle - 1;  
 }  
 middle = (first + last)/2;  
 }  
 if ( first > last )  
 System.out.println(item + " is not found.\n");  
 }  
}

Output 1:

Enter number of elements:  
7  
Enter 7 integers  
4  
5  
66  
77  
8  
99  
0  
Enter the search value:  
77  
77 found at location 4.

Output 2:

Enter number of elements:  
5  
Enter 5 integers  
12  
3  
77  
890  
23  
Enter the search value:  
99  
99 is not found.

-------------------------------------------------------------------------------------

Java program to generate random number – Example

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: APRIL 21, 2014

Example Program to generate random numbers

In the below program, we are using the nextInt() method of Randomclass to serve our purpose.

/\* Program: Random number generator  
 \* Written by: Chaitanya from beginnersbook.com  
 \* Input: None  
 \* Output:Random number between o and 200\*/  
import java.util.\*;  
class GenerateRandomNumber {  
 public static void main(String[] args) {  
 int counter;  
 Random rnum = new Random();  
 /\* Below code would generate 5 random numbers  
 \* between 0 and 200.  
 \*/  
 System.out.println("Random Numbers:");  
 System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 for (counter = 1; counter <= 5; counter++) {  
 System.out.println(rnum.nextInt(200));  
 }  
 }  
}

Output:

Random Numbers:  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
135  
173  
5  
17  
15

The output of above program would not be same everytime. It would generate any 5 random numbers between 0 and 200 whenever you run this code. For e.g. When I ran it second time, it gave me the below output, which is entirely different from the above one.

Output 2:

Random Numbers:  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
46  
99  
191  
7  
134

-------------------------------------------------------------------------------------

Java Program to find duplicate Characters in a String

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JULY 15, 2014

This program would find out the duplicate characters in a String and would display the count of them.

import java.util.HashMap;  
import java.util.Map;  
import java.util.Set;  
   
public class Details {  
   
 public void countDupChars(String str){  
   
 //Create a HashMap   
 Map<Character, Integer> map = new HashMap<Character, Integer>();   
   
 //Convert the String to char array  
 char[] chars = str.toCharArray();  
   
 /\* logic: char are inserted as keys and their count  
 \* as values. If map contains the char already then  
 \* increase the value by 1  
 \*/  
 for(Character ch:chars){  
 if(map.containsKey(ch)){  
 map.put(ch, map.get(ch)+1);  
 } else {  
 map.put(ch, 1);  
 }  
 }  
   
 //Obtaining set of keys  
 Set<Character> keys = map.keySet();  
   
 /\* Display count of chars if it is  
 \* greater than 1. All duplicate chars would be   
 \* having value greater than 1.  
 \*/  
 for(Character ch:keys){  
 if(map.get(ch) > 1){  
 System.out.println("Char "+ch+" "+map.get(ch));  
 }  
 }  
 }  
   
 public static void main(String a[]){  
 Details obj = new Details();  
 System.out.println("String: BeginnersBook.com");  
 System.out.println("-------------------------");  
 obj.countDupChars("BeginnersBook.com");  
   
 System.out.println("\nString: ChaitanyaSingh");  
 System.out.println("-------------------------");  
 obj.countDupChars("ChaitanyaSingh");  
   
 System.out.println("\nString: #@$@!#$%!!%@");  
 System.out.println("-------------------------");  
 obj.countDupChars("#@$@!#$%!!%@");  
 }  
}

Output:

String: BeginnersBook.com  
-------------------------  
Char e 2  
Char B 2  
Char n 2  
Char o 3  
  
String: ChaitanyaSingh  
-------------------------  
Char a 3  
Char n 2  
Char h 2  
Char i 2  
  
String: #@$@!#$%!!%@  
-------------------------  
Char # 2  
Char ! 3  
Char @ 3  
Char $ 2  
Char % 2

Reference:

[**HashMap**](http://beginnersbook.com/2013/12/hashmap-in-java-with-example/)

-------------------------------------------------------------------------------------

Java program for binary to decimal conversion

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JULY 20, 2014

There are two following ways to convert binary number to decimal number:

1) Using [**Integer.parseInt() method**](http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html#parseInt(java.lang.String,%20int)) of Integer class.

2) Do conversion by writing your own logic without using any predefined methods.

Method 1: Binary to Decimal conversion using Integer.parseInt() method

import java.util.Scanner;  
class BinaryToDecimal {  
 public static void main(String args[]){  
 Scanner input = new Scanner( System.in );  
 System.out.print("Enter a binary number: ");  
 String binaryString =input.nextLine();  
 System.out.println("Output: "+Integer.parseInt(binaryString,2));  
 }  
}

Output:

Enter a binary number: 1101  
Output: 13

Method 2: Conversion without using parseInt

public class Details {  
   
 public int BinaryToDecimal(int binaryNumber){  
   
 int decimal = 0;  
 int p = 0;  
 while(true){  
 if(binaryNumber == 0){  
 break;  
 } else {  
 int temp = binaryNumber%10;  
 decimal += temp\*Math.pow(2, p);  
 binaryNumber = binaryNumber/10;  
 p++;  
 }  
 }  
 return decimal;  
 }  
   
 public static void main(String args[]){  
 Details obj = new Details();  
 System.out.println("110 --> "+obj.BinaryToDecimal(110));  
 System.out.println("1101 --> "+obj.BinaryToDecimal(1101));  
 System.out.println("100 --> "+obj.BinaryToDecimal(100));  
 System.out.println("110111 --> "+obj.BinaryToDecimal(110111));  
 }  
}

Output:

110 --> 6  
1101 --> 13  
100 --> 4  
110111 --> 55

-------------------------------------------------------------------------------------

Java Program to get input from user

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JULY 16, 2014

In this tutorial we are gonna see how to accept input from user. We are using Scanner class to get the input. In the below example we are getting input String, integer and a float number. For this we are using following methods:

1) public String nextLine(): For getting input String

2) public int nextInt(): For integer input

3) public float nextFloat(): For float input

Example:

import java.util.Scanner;  
  
class GetInputData  
{  
 public static void main(String args[])  
 {  
 int num;  
 float fnum;  
 String str;  
   
 Scanner in = new Scanner(System.in);  
   
 //Get input String  
 System.out.println("Enter a string: ");  
 str = in.nextLine();  
 System.out.println("Input String is: "+str);  
   
 //Get input Integer  
 System.out.println("Enter an integer: ");  
 num = in.nextInt();  
 System.out.println("Input Integer is: "+num);  
   
 //Get input float number  
 System.out.println("Enter a float number: ");  
 fnum = in.nextFloat();  
 System.out.println("Input Float number is: "+fnum);   
 }  
}

Output:

Enter a string:   
Chaitanya  
Input String is: Chaitanya  
Enter an integer:   
27  
Input Integer is: 27  
Enter a float number:   
12.56  
Input Float number is: 12.56

Reference:

[**Scanner Javadoc**](http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html)

Java program to convert decimal to binary

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JULY 16, 2014

There are three following ways to convert Decimal number to binary number:

1) Using [**toBinaryString() method**](http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html#toBinaryString(int)) of Integer class.

2) Do conversion by writing your own logic without using any predefined methods.

3) Using Stack

Method 1: Using toBinaryString() method

class DecimalBinaryExample{  
   
 public static void main(String a[]){  
 System.out.println("Binary representation of 124: ");  
 System.out.println(Integer.toBinaryString(124));  
 System.out.println("\nBinary representation of 45: ");  
 System.out.println(Integer.toBinaryString(45));  
 System.out.println("\nBinary representation of 999: ");  
 System.out.println(Integer.toBinaryString(999));  
 }  
}

Output:

Binary representation of 124:   
1111100  
  
Binary representation of 45:   
101101  
  
Binary representation of 999:   
1111100111

Method 2: 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);  
 }  
}

Output:

Binary representation of 124:   
1111100  
Binary representation of 45:   
101101  
Binary representation of 999:   
1111100111

Method 3: 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();  
 }  
}

Output:

Enter decimal number:   
999  
  
Binary representation is:1111100111

java program to find factorial of a given number using recursion

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JANUARY 18, 2014

Here we will write programs to find out the factorial of a number using recursion.

Program 1:

Program will prompt user for the input number. Once user provide the input, the program will calculate the factorial for the provided input number.

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: User would enter the 10 elements  
 \* and the program will store them into an array and   
 \* will display the sum of them.  
 \*/  
import java.util.Scanner;  
class FactorialDemo{  
 public static void main(String args[]){  
 //Scanner object for capturing the user input  
 Scanner scanner = new Scanner(System.in);  
 System.out.println("Enter the number:");  
 //Stored the entered value in variable  
 int num = scanner.nextInt();  
 //Called the user defined function fact  
 int factorial = fact(num);  
 System.out.println("Factorial of entered number is: "+factorial);  
 }  
 static int fact(int n)  
 {  
 int output;  
 if(n==1){  
 return 1;  
 }  
 //Recursion: Function calling itself!!  
 output = fact(n-1)\* n;  
 return output;  
 }  
}

Output:

Enter the number:  
5  
Factorial of entered number is: 120

Program 2:

If you do not want user intervention and simply want to specify the number in program itself then refer this example.

class FactorialDemo2{  
 public static void main(String args[]){  
 int factorial = fact(4);  
 System.out.println("Factorial of 4 is: "+factorial);  
 }  
 static int fact(int n)  
 {  
 int output;  
 if(n==1){  
 return 1;  
 }  
 //Recursion: Function calling itself!!  
 output = fact(n-1)\* n;  
 return output;  
 }  
}

Output:

Factorial of 4 is: 24

Java Program to find duplicate Characters in a String

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JULY 15, 2014

This program would find out the duplicate characters in a String and would display the count of them.

import java.util.HashMap;  
import java.util.Map;  
import java.util.Set;  
   
public class Details {  
   
 public void countDupChars(String str){  
   
 //Create a HashMap   
 Map<Character, Integer> map = new HashMap<Character, Integer>();   
   
 //Convert the String to char array  
 char[] chars = str.toCharArray();  
   
 /\* logic: char are inserted as keys and their count  
 \* as values. If map contains the char already then  
 \* increase the value by 1  
 \*/  
 for(Character ch:chars){  
 if(map.containsKey(ch)){  
 map.put(ch, map.get(ch)+1);  
 } else {  
 map.put(ch, 1);  
 }  
 }  
   
 //Obtaining set of keys  
 Set<Character> keys = map.keySet();  
   
 /\* Display count of chars if it is  
 \* greater than 1. All duplicate chars would be   
 \* having value greater than 1.  
 \*/  
 for(Character ch:keys){  
 if(map.get(ch) > 1){  
 System.out.println("Char "+ch+" "+map.get(ch));  
 }  
 }  
 }  
   
 public static void main(String a[]){  
 Details obj = new Details();  
 System.out.println("String: BeginnersBook.com");  
 System.out.println("-------------------------");  
 obj.countDupChars("BeginnersBook.com");  
   
 System.out.println("\nString: ChaitanyaSingh");  
 System.out.println("-------------------------");  
 obj.countDupChars("ChaitanyaSingh");  
   
 System.out.println("\nString: #@$@!#$%!!%@");  
 System.out.println("-------------------------");  
 obj.countDupChars("#@$@!#$%!!%@");  
 }  
}

Output:

String: BeginnersBook.com  
-------------------------  
Char e 2  
Char B 2  
Char n 2  
Char o 3  
  
String: ChaitanyaSingh  
-------------------------  
Char a 3  
Char n 2  
Char h 2  
Char i 2  
  
String: #@$@!#$%!!%@  
-------------------------  
Char # 2  
Char ! 3  
Char @ 3  
Char $ 2  
Char % 2

Reference:

[**HashMap**](http://beginnersbook.com/2013/12/hashmap-in-java-with-example/)

Java program to sum the elements of an array

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JANUARY 18, 2014

In this tutorial we will see how to sum up all the elements of an array.

Program 1: No user interaction

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: Get sum of array elements  
 \*/  
class SumOfArray{  
 public static void main(String args[]){  
 int[] array = {10, 20, 30, 40, 50, 10};  
 int sum = 0;  
 //Advanced for loop  
 for( int num : array) {  
 sum = sum+num;  
 }  
 System.out.println("Sum of array elements is:"+sum);  
 }  
}

Output:

Sum of array elements is:160

Program 2: User enters the array’s elements

/\*\*  
 \* @author: BeginnersBook.com  
 \* @description: User would enter the 10 elements  
 \* and the program will store them into an array and   
 \* will display the sum of them.  
 \*/  
import java.util.Scanner;  
class SumDemo{  
 public static void main(String args[]){  
 Scanner scanner = new Scanner(System.in);  
 int[] array = new int[10];  
 int sum = 0;  
 System.out.println("Enter the elements:");  
 for (int i=0; i<10; i++)  
 {  
 array[i] = scanner.nextInt();  
 }  
 for( int num : array) {  
 sum = sum+num;  
 }  
 System.out.println("Sum of array elements is:"+sum);  
 }  
}

Output:

Enter the elements:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Sum of array elements is:55

Java program for bubble sort in Ascending & descending order

CHAITANYA SINGH | FILED UNDER: [**JAVA EXAMPLES**](http://beginnersbook.com/category/java-examples/) | UPDATED: JULY 16, 2014

In this tutorial we are gonna see how to do sorting in ascending & descending order using [**Bubble sort**](http://en.wikipedia.org/wiki/Bubble_sort) algorithm.

Bubble sort program for sorting in ascending Order

import java.util.Scanner;  
  
class BubbleSortExample {  
 public static void main(String []args) {  
 int num, i, j, temp;  
 Scanner input = new Scanner(System.in);  
   
 System.out.println("Enter the number of integers to sort:");  
 num = input.nextInt();  
   
 int array[] = new int[num];  
   
 System.out.println("Enter " + num + " integers: ");  
   
 for (i = 0; i < num; i++)   
 array[i] = input.nextInt();  
   
 for (i = 0; i < ( num - 1 ); i++) {  
 for (j = 0; j < num - i - 1; j++) {  
 if (array[j] > array[j+1])   
 {  
 temp = array[j];  
 array[j] = array[j+1];  
 array[j+1] = temp;  
 }  
 }  
 }  
   
 System.out.println("Sorted list of integers:");  
   
 for (i = 0; i < num; i++)   
 System.out.println(array[i]);  
 }  
}

Output:

Enter the number of integers to sort:  
6  
Enter 6 integers:   
12  
6  
78  
9  
45  
08  
Sorted list of integers:  
6  
8  
9  
12  
45  
78

Bubble sort program for sorting in descending Order

In order to sort in descending order we just need to change the logic array[j] > array[j+1] to array[j] < array[j+1] in the above program. Complete code as follows:

import java.util.Scanner;  
  
class BubbleSortExample {  
 public static void main(String []args) {  
 int num, i, j, temp;  
 Scanner input = new Scanner(System.in);  
   
 System.out.println("Enter the number of integers to sort:");  
 num = input.nextInt();  
   
 int array[] = new int[num];  
   
 System.out.println("Enter " + num + " integers: ");  
   
 for (i = 0; i < num; i++)   
 array[i] = input.nextInt();  
   
 for (i = 0; i < ( num - 1 ); i++) {  
 for (j = 0; j < num - i - 1; j++) {  
 if (array[j] < array[j+1])   
 {  
 temp = array[j];  
 array[j] = array[j+1];  
 array[j+1] = temp;  
 }  
 }  
 }  
   
 System.out.println("Sorted list of integers:");  
   
 for (i = 0; i < num; i++)   
 System.out.println(array[i]);  
 }   
}

Output:

Enter the number of integers to sort:  
6  
Enter 6 integers:   
89  
12  
45  
9  
56  
102  
Sorted list of integers:  
102  
89  
56  
45  
12  
9

Reverse an array without affecting special characters

Given a string, that contains special character together with alphabets (‘a’ to ‘z’ and ‘A’ to ‘Z’), reverse the string in a way that special characters are not affected.

Examples:

Input: str = "a,b$c"  
Output: str = "c,b$a"  
Note that $ and , are not moved anywhere.   
Only subsequence "abc" is reversed  
  
Input: str = "Ab,c,de!$"  
Output: str = "ed,c,bA!$"

**We strongly recommend you to minimize your browser and try this yourself first.**

**Simple Solution:**

1) Create a temporary character array say temp[].

2) Copy alphabetic characters from given array to temp[].

3) Reverse temp[] using standard [string reversal algorithm](http://www.geeksforgeeks.org/write-a-program-to-reverse-an-array-or-string/).

4) Now traverse input string and temp in a single loop. Wherever there is alphabetic character is input string, replace it with current character of temp[].

**Efficient Solution:**

Time complexity of above solution is O(n), but it requires extra space and it does two traversals of input string.

We can reverse with one traversal and without extra space. Below is algorithm.

1) Let input string be 'str[]' and length of string be 'n'  
2) l = 0, r = n-1  
3) While l is smaller than r, do following  
 a) If str[l] is not an alphabetic character, do l++  
 b) Else If str[r] is not an alphabetic character, do r--  
 c) Else swap str[l] and str[r]

Queries for characters in a repeated string

Given a string **X**. Form a string S by repeating string X multiple times i.e appending string X multiple times with itself. There are **Q** queries of form i and j. The task is to print “Yes” if the element at index i is same as the element at index j in S else print “No” for each query.

Examples:

Input : X = "geeksforgeeks", Q = 3.  
Query 1: 0 8  
Query 2: 8 13  
Query 3: 6 15  
  
Output :  
Yes  
Yes  
No  
  
String S will be "geeksforgeeksgeeksforgeeks....".  
For Query 1, index 0 and index 8 have same element i.e 'g'.  
For Query 2, index 8 and index 13 have same element i.e 'g'.  
For Query 3, index 6 = 'o' and index 15 = 'e' which are not same.

[We strongly recommend that you click here and code it yourself first, before moving on to the solution.](http://code.geeksforgeeks.org/)

Let length of string X be n. Observe that element at indexes 0, n, 2n, 3n,…. are same. Similarly for index i, position i, n+i, 2n+i, 3n+i,….. contain same element.

So, for each query, find (i%n) and (j%n) and if both are same for string X

Program to toggle all characters in a string

in a given string in which we toggle all characters.

Examples:

Input : gfg  
Output : GFG  
  
Input : aBc12#  
Output : AbC12#  
  
Input : tu@kmiNi  
Output : TU@KMInI

[We strongly recommend that you click here and code it yourself first, before moving on to the solution.](http://code.geeksforgeeks.org/)

Traverse the given string, if uppercase characters comes, convert into lowercase and lowercase latter convert into uppercase.

|  |
| --- |
| // c++ program to toggle all characters  #include<bits/stdc++.h>  using namespace std;    void toggleChars(char str[])  {  for (int i=0; str[i]!='\0'; i++)  {  if (str[i]>='A' && str[i]<='Z')  str[i] = str[i] + 'a' - 'A';  else if (str[i]>='a' && str[i]<='z')  str[i] = str[i] + 'A' - 'a';  }  }    // Driver code  int main()  {  char str[] = "GeKf@rGeek$";  toggleChars(str);  cout << "String after toggle " << endl;  cout << str << endl;  return 0;  } |

Run on IDE

Output:

gEkF@RgEEK$

This article is contributed by **MATHE\_KA\_BANDA**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Check whether Strings are k distance apart or not

Given two strings, the task is to find if they are only less than or equal to k edit distance apart. It means that strings are only k edit distance apart when there are only k mismatches.

Print Yes if there are less than or equal to k mismatches, Else No.

Also print yes if both strings are already same.

Examples:

Input : str1 = "geeksforgeeks"   
 str2 = "geeksforgeek"   
 k = 1  
Output : Yes  
Explanation: Only one character is mismatched   
 or to be removed i.e. s at last   
  
Input : str1 = "nishant"   
 str2 = "nisha"   
 k = 1  
Output : No   
Explanation: 2 characters need to be removed  
 i.e. n and t   
  
Input : str1 = "practice"   
 str2 = "prac"   
 k = 3  
Output : No  
Explanation: 4 characters are mismatched or to  
 be removed i.e. t,i,c,e at last i.e. > k  
  
Input : str1 = "Ping" str2 = "Paging" k = 2  
Output : Yes   
Explanation: 2 characters need to be removed or   
 mismatched i.e. a and g in paging

Queries on substring palindrome formation

Given a string **S**, and two type of queries.

Type 1: 1 **L x**, Indicates update Lth index   
 of string S by x character.  
Type 2: 2 **L R**, Find if characters between position L and R   
 of string S can form a palindrome string.   
 If palindrome can be formed print "Yes",   
 else print "No".  
1 <= L, R <= |S|

Examples:

Input : S = "geeksforgeeks"  
Query 1: 1 4 g  
Query 2: 2 1 4  
Query 3: 2 2 3  
Query 4: 1 10 t  
Query 5: 2 10 11  
Output :  
Yes  
Yes  
No  
  
Query 1: update index 3 (position 4) of string S by   
character 'g'. So new string S = "geegsforgeeks".  
  
Query 2: find if rearrangement between index 0 and 3  
can form a palindrome. "geegs" is palindrome, print "Yes".  
  
Query 3: find if rearrangement between index 1 and 2   
can form a palindrome. "ee" is palindrome, print "Yes".  
  
Query 4: update index 9 (position 10) of string S by   
character 't'. So new string S = "geegsforgteks".  
  
Query 3: find if rearrangement between index 9 and 10   
can form a palindrome. "te" is not palindrome, print "No".

[We strongly recommend that you click here and code it yourself first, before moving on to the solution.](http://code.geeksforgeeks.org/)

Substring S[L…R] form a palindrome only if frequencies of all the character in S[L…R] are even, with one except allowed.

For query of type 1, simply update string   
S[L] by character x.  
  
For each query of type 2, calculate the   
frequency of character and check if   
frequencies of all characters is even (with)  
one exception allowed.

Following are two different methods to find frequency of each character in S[L…R]:

**Method 1: Use a frequency array to find the frequency of each element in S[L…R].**

Below is C++ implementation of this approach:

Queries on subsequence of string

Given a string **S** and **Q** queries, each query contains a string **T**. The task is print “Yes” if T is subsequence of S, else print “No”.

Examples:

Input : S = "geeksforgeeks"  
Query 1: "gg"  
Query 2: "gro"  
Query 3: "gfg"  
Query 4: "orf"  
  
Output :  
Yes  
No  
Yes  
No

[We strongly recommend that you click here and code it yourself first, before moving on to the solution.](http://code.geeksforgeeks.org/)

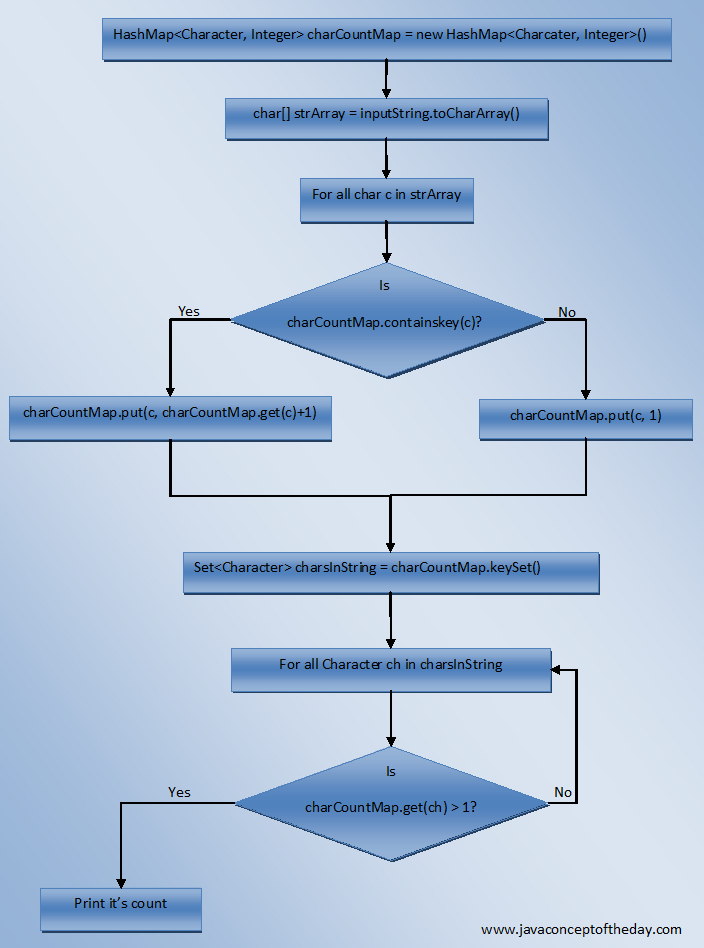
For each query, using the **brute force**, start iterating over S looking for the first character of T. As soon as, the first character is found, continue to iterate S now looking for the second character of T and so on (Refer [this](http://www.geeksforgeeks.org/given-two-strings-find-first-string-subsequence-second/) for details). If manage to find all the character of T, print “Yes”, else “No”. Time complexity is be O(Q\*N), N is the length of S.

The **efficient approach** can be if we know the position of next character of T in S. Then simply skip all the character between current and position of next character and jump to that position. This can be done by making |S| x 26 size matrix and storing the next position of each character from every position of S.

Logic Used To Find Duplicate Characters In A String In Java :

We use HashMap and Set to find the duplicate characters in a string. First, we convert the given string to char array. We then create one HashMap with Character as a key and it’s number of occurrences as a value. Then we extract a Set containing all keys of this HashMap using keySet() method. Then we use this keySet to get the duplicate characters i.e characters which have appeared more than once in the given string.

Flowchart :



5 Difference between Iterator and ListIterator in Java?

The Iterator is the standard way to traverse a collection in Java. You can use Iterator to traverse a List, Set, Map, Stack, Queue or any Collection, but you might not know that there is another way to traverse over List in Java? Yes, it's called the ListIterator. There are many differences between Iterator and ListIterator in Java, but the most significant of them is that [Iterator](http://www.java67.com/2013/02/java-iterator-example.html) only allows you to traverse in **one direction** i.e. forward, you have just got a next() method to get the next element, there is no previous() method to get the previous element. On the other hand, ListIterator allows you to traverse the list in **both directions** i.e. forward and backward. It has got both next()and previous() method to access the next and previous element from List.

Unfortunately, ListIterator only supports the [List](http://www.java67.com/2012/12/how-to-create-and-initialize-list-arraylist-same-line.html) interface, you cannot use it to traverse over [Map](http://www.java67.com/2015/01/how-to-convert-map-to-list-in-java.html) or [Set](http://www.java67.com/2014/01/how-hashset-is-implemented-or-works-internally-java.html) in Java. That was the basic *difference between Iterator and ListIterator* class, let's see a couple of more to answer this question in detail on the Java interviews.

Iterator vs ListIterator

You can differentiate between Iterator and ListIterator on following topics :

* Direction of traversal
* Operation allowed during iteration
* Supported Collection classes
* Iterating from any arbitrary element
* Supported methods

Let's see each of them in little bit detail.

**Traversal direction**

As I told you in the first paragraph that fundamental difference between Iterator and ListIterator is that former allow traversal only in **forward direction** i.e. you cannot go back once you move forward, which is what sometimes you need. ListIterator gives you that functionality via previous() method. Similar to next() this method returns the previous element hence support traversal in the backward direction (see [Core Java Volume 1 - Fundamentals](https://www.amazon.com/Core-Java-I--Fundamentals-10th/dp/0134177304?tag=javamysqlanta-20) by Cay. S. Horstmann)

Read more: <http://www.java67.com/2016/09/5-difference-between-iterator-and-ListIterator-in-java.html#ixzz4YYcUVQT6>

Reversible numbers

A number is said to be a reversible if sum of the number and its reverse had only odd digits. The problem is to find out if a number is reversible or not.

Examples:

**Input:** 36   
**Output:** Reversible number  
as 36 + 63 = 99 has only odd digits.  
  
**Input:** 409   
**Output:** Reversible number  
as 409 + 904 = 1313 has only odd digits.  
  
**Input:** 35   
**Output:** Not Reversible number  
as 35 + 53 = 88 has only odd digits

**Naive Method**

Calculate reverse of each number and add it to the number. If the resultant is reversible increment the value of count. Calculate this for every number from 1 to n.

**Time complexity:** O(10^n) as it should calculate reverse of each number.

**Advance method**

* **1 digit number:** Any one digit number will add to itself, which always be an even number, And thus there are no solutions.
* **2 digits number:** Both digits must be odd.
  + If a+b > 10 ,then we have a carryover and thus the first digit of the result will have a different parity than the second digit.
  + So, Solutions can only be formed where a+b < 10 and a + b is odd. So, total 20 such numbers are possible.
* **3 digits number:**
  + The middle digit needs to be added to itself. That means that the third digit must have a carryover and be odd.
  + Since the third digit is odd the first digit is odd as well if the second digit does not have a carryover, which happens when the second digit is less than 5, which gives us 20 choices for the first/third digit set and 5 options for the middle.So, totals 100 pairs.
* **4 digits number:** There are two pairs, say the inner and outer pair.
  + If the inner pair has carryover then the outer pair must also have carryover.
  + Otherwise, the two inner pairs will have different parity.
  + If the inner pair has carryover then the outer pair will have different parity since the first digit will end up with a carry over which the last digit would not get.
  + Therefore we have solutions only when none of the pairs have carry over.
  + In total: For the outer pair, this gives us the 20 choices we have seen in the two digit case. And it gives us 30 cases for the inner pair since they can also contain a zero.
  + Or in total we get 20\*30 = 600 solutions.
* **5, 9, 13.. digits number:** No solution as 1-digit number.
* **6, 8, 10.. digits number:** Same as 2-digits number i.e. if n = 2\*k then total solution = 20\*30^(k-1).
* **7, 11, 15.. digits number:** Same as 3-digits number i.e if n = 4k + 3 then total solution = 100\*500^(k).

**Generalizing the Solution:**

* All even numbered digits(2, 4, 6, 8..) have the same formula so we can generalize
* that for some integer k such that n = 2k we have 20\*30^(k-1) solutions
* which represents the outer pair along with all the inner pairs.
* For n (3, 7, 11..) of form 4k + 3 (k is an integer), we have that the middle digit and
* the outer pair gives us 5 and 20 options, as in the case of 3 digit number.
* Then we have sets of internal pairs which gives us 20 and 25 solutions.
* So that means we can generalize it to 20\*5\*(20\*25)^(k) = 100\*500^(k).
* For n of form 4k+ 1 which means 1, 5, 9.. none of these have any solutions.