
//In Lab 1: Using recursion, write a program that takes a word as an input and outputs that word backwards.

```
import java.util.Scanner;

public class ReverseWord
{
    public static void main( String [] args )
    {

        Scanner scan = new Scanner( System.in );
        System.out.print( "Enter a word > " );
        String word = scan.next( );

        String reversedWord = reverseWordRecursive( word );
        System.out.println( word + " reversed is " + reversedWord );
    }

    public static String reverseWordRecursive( String s )
    {
        if( s.length( ) == 0 ) // base case
            return "";
        else // general case
            return( s.charAt( s.length( ) - 1 )
                + reverseWordRecursive( s.substring( 0, s.length( ) - 1 ) ) );
    }
}
```

```

//In_Lab 2
/** RecursiveBinarySearch class
 * Searching a sorted array (descending order) using recursion
 * Anderson, Franceschi
 */

import java.util.Scanner;

public class RecursiveBinarySearch
{
    public static void main( String [] args )
    {
        Scanner scan = new Scanner( System.in );

        int [] numbers = { 95, 89, 72, 69, 64, 53, 51, 48, 45,
                           36, 22, 15, 12, 8, 7, 6, 3 };

        int value = 0;

        // prompt user and read value
        System.out.print( "Enter an integer to find > " );
        value = scan.nextInt( );

        int index = binarySearchRecursive( numbers, value, 0, numbers.length - 1 );
        if ( index != - 1 )
            System.out.println( value + " found at index " + index );
        else
            System.out.println( value + " not found" );
    }

    public static int binarySearchRecursive( int [ ] arr, int key, int start, int end )
    {
        if ( start <= end )
        {
            int middle = ( start + end ) / 2;
            if ( arr[middle] == key )    // found key, base case
                return middle;
            else if ( arr[middle] < key ) // look lower
                return binarySearchRecursive( arr, key, start, middle - 1 );
            else                          // look higher
                return binarySearchRecursive( arr, key, middle + 1, end );
        }
        else                          // key not found, base case
            return -1;
    }
}

```

```
/** InClass_3: Decimal class
 * Recursive method that converts int to a String representing that
 */ number in binary. It needs a client.
```

```
public class Decimal
{
    private int number;

    /** Constructor
     * @param newNumber the number
     */
    public Decimal( int newNumber )
    {
        setNumber( newNumber );
    }

    /** getNumber method
     * @return the number
     */
    public int getNumber( )
    {
        return number;
    }

    /**
     * Mutator method:<BR>
     * Allows client to set values of number
     * <B>setNumber</B> sets the value
     * in <B>number</B> to the absolute value of newNumber
     * @param number the new int for number
     */
    public void setNumber( int number )
    {
        this.number = Math.abs( number );
    }

    /** toString
     * @return the number as a String
     */
    public String toString( )
    {
        return ( String.valueOf( number ) );
    }

    /** equals
     * @param o Decimal object
     * @return return true if the number in d is equal to
     * the number in this object
     */
    public boolean equals( Object o )
    {
        if ( ! ( o instanceof Decimal ) )
```

```

        return false;
    else
    {
        Decimal d = (Decimal) o;
        return ( number == d.number );
    }
}

/**
 * convertToBinary method
 * converts number to its binary equivalent
 * @return returns a String, the binary equivalent of number
 */
public String convertToBinary( )
{
    return convertToBinary( number );
}

/**
 * private convertToDecimal method
 * converts number to its decimal equivalent
 * @param n String representing a binary number
 * @return returns an int, the decimal equivalent of n
 */
private String convertToBinary( int n )
{
    if ( n / 2 == 0 ) // n is 0 or 1
        return ( String.valueOf( n % 2 ) );
    else
        return ( convertToBinary( n / 2 ) + ( n % 2 ) );
}
}

-----
/** New File Here DecimalClient class
 * Input an integer, convert it to binary
 * Anderson, Franceschi
 */

import java.util.Scanner;

public class DecimalClient
{
    public static final int ZERO = 0;

    public static void main( String [] args )
    {
        Scanner scan = new Scanner( System.in );
        int decimal = 0;

        boolean validInt = false;
        String message = "";
        do
        {
            // prompt user and read value

```

```
System.out.print( message + "Enter an integer greater than or equal to 0 > " );
decimal = scan.nextInt( );

if ( decimal < ZERO )
    message = "Integer must be greater than or equal to " + ZERO + "\n";
else
    validInt = true;

} while ( !validInt );

Decimal d = new Decimal( decimal );
String binary = d.convertToBinary( );
System.out.println( decimal + " in binary is " + binary );
}
}
```