package lab\_3;

import java.util.Arrays;//import the Arrays object

public class Lab\_3 {

public static int sumOfDigits(int number){

if (number < 10) //base case

return number;

else{

return (number % 10) + sumOfDigits(number / 10); //recursive call

}//else

}//sumOfDigits

public static int recursiveBinarySearch(int[] array, int key){

int lowindex = 0;

int highindex = array.length-1;

return binarySearchRecursive(array,key,lowindex,highindex);

}//recursiveBinarySearch

public static int binarySearchRecursive(int[] array, int key, int lowindex, int highindex){

if(lowindex > highindex)

return -1;

int mid = ((lowindex + highindex) / 2);

if(key == array[mid])

return mid;

else if(array[mid] > key){

return binarySearchRecursive(array, key,lowindex,mid-1) ;

}//if

else{

return binarySearchRecursive(array, key, mid + 1, highindex);

}//else if

}//binarySearchRecursive

public static int exponent(int num, int exp){

if (exp == 0) return 1;//base case

else if (exp % 2 == 0 ){ //if number is even

return(exponent(num,exp/2) \* exponent(num,exp/2)); //even number break

}//else if

else

return(num \* exponent(num, exp-1)); //odd number

}//exponent

public static boolean isBackwards(String first, String second) throws IllegalArgumentException {

first = first.toLowerCase(); //method to bring the first and second string to lowercase strings

second = second.toLowerCase();

if ((first == null)||(second == null)){ //base case to throw an error

System.out.println("Error"); //display

throw new IllegalArgumentException();}//if

System.out.println("First = " + first + " / Second = " + second);

//Sout Echo

if (first.length() != second.length()){//Compare the two Strings

return false; //false

}//if

if (first.length() == 0 && second.length() == 0){ //if both lengths are zero

return true; //Base case

}//if

String compare1 = "" + first.charAt(first.length() - 1); //

String compare2 = "" + second.charAt(0);

if (compare1.equalsIgnoreCase(compare2)){

//check 1st of first and last of 2nd

//if they are equal then the next subsequent characters are check for consistency

String temp1 = first.substring(0, first.length() - 1);//move to next postition

String temp2 = second.substring(1, second.length());//move to next position

return isBackwards(temp1, temp2); //recursive call to check for the remaining characters are palidrones

}//if

return false;//the end

}//isBackwards

public static void main(String[] args){

//// Test Number 1

System.out.println(sumOfDigits(734));

//// Test Number 2

//initialize an unsorted array

int[] arr = new int[]{9,34,2,15,90,56,44,78,12,23,86};

Arrays.sort(arr); //use arrays sort method in the arrays class

int key = 34; // search for this key

int position =binarySearchRecursive(arr, key, 0, arr.length-1);

if(position != -1){

System.out.println("Found search key " + key + " at index " + position);

System.out.println("Search key not found");

}//if

System.out.println(exponent(3,0));

System.out.println(exponent(3,2));

System.out.println(exponent(9,0));

// Testing the isBackwards method

System.out.println("Testing isBackwards");

System.out.println("Problem 4");

System.out.println("Fried and deirf");

System.out.println("Should return true = " + isBackwards("Fried","deirf"));

System.out.println("Nascar and racsaN");

System.out.println("Should be true = " + isBackwards("Nascar","racsaN"));

System.out.println("Pacsun and nuscaP");

System.out.println("Should be true = " + isBackwards("Pacsun","nuscaP"));

System.out.println("Sit and Toes");

System.out.println("Should return false = " + isBackwards("Sit","Toes"));

System.out.println("Blank strings ");

System.out.println("should return true = " + isBackwards("",""));

System.out.println("I Madam, and mad am I");

System.out.println("Should be true = " + isBackwards("I madam","mad am I"));

}//main

}//class

Output:

=======================================================================

run:

14

Found search key 34 at index 5

Search key not found

1

9

1

Testing isBackwards

Problem 4

Fried and deirf

First = fried / Second = deirf

First = frie / Second = eirf

First = fri / Second = irf

First = fr / Second = rf

First = f / Second = f

First = / Second =

Should return true = true

Nascar and racsaN

First = nascar / Second = racsan

First = nasca / Second = acsan

First = nasc / Second = csan

First = nas / Second = san

First = na / Second = an

First = n / Second = n

First = / Second =

Should be true = true

Pacsun and nuscaP

First = pacsun / Second = nuscap

First = pacsu / Second = uscap

First = pacs / Second = scap

First = pac / Second = cap

First = pa / Second = ap

First = p / Second = p

First = / Second =

Should be true = true

Sit and Toes

First = sit / Second = toes

Should return false = false

Blank strings

First = / Second =

should return true = true

I Madam, and mad am I

First = i madam / Second = mad am i

Should be true = false

BUILD SUCCESSFUL (total time: 0 seconds)

