Practical No 3

Extended Euclid

Input:-

package exteuclid;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class ExtEuclid {

public static void main(String[] args) throws IOException {

int [] ans = new int[3];

int x,y,a,b;

BufferedReader br = new BufferedReader (new InputStreamReader(System.in));

System.out.println("Enter the first non negative number :");

a=Integer.parseInt(br.readLine());

System.out.println("Enter the second non negative number:");

b=Integer.parseInt(br.readLine());

ans=Euclid(a,b);

System.out.println("GCD of" +a+ "and" +b+ ";");

System.out.println("\n gcd ("+a+","+b+")="+ans[0]+"\n");

System.out.println("Extended form :\n");

//System.out.println(" "+ans[0]+" "+ans[1]+" "+ans[2]+"");

//System.out.println(" "+a+" "+b+"\n");

System.out.println(" d="+ans[0]+" s="+ans[1]+" t="+ans[2]+"");

}

public static int[]Euclid(int a, int b)

{

int[]ans=new int[3];

int q;

if(b==0)

{

ans[0]=a;

ans[1]=1;

ans[2]=0;

}

else{

q=a/b;

ans=Euclid(b,a%b);

int temp=ans[1]-ans[2]\*q;

ans[1]=ans[2];

ans[2]=temp;

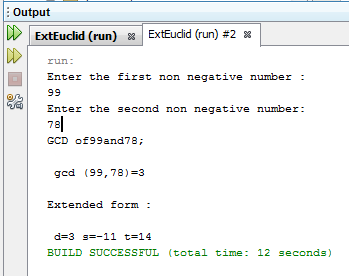
}

return ans;

}

}

Ouput:



Practical no 4

Mergeort

Input:

package mergesort;

import java.util.Scanner;

public class MergeSort {

public static void sort(int[] a, int low, int high){

int N = high - low;

if (N <= 1)

return;

int mid = low + N/2;

sort(a, low, mid);

sort(a, mid, high);

int[] temp = new int[N];

int i = low, j = mid;

for (int k = 0; k < N; k++){

if (i == mid)

temp[k] = a[j++];

else if (j == high)

temp[k] = a[i++];

else if (a[j]<a[i])

temp[k] = a[j++];

else

temp[k] = a[i++];

}

for (int k = 0; k < N; k++)

a[low + k] = temp[k];

}

public static void main(String[] args) {

Scanner scan = new Scanner( System.in );

System.out.println("Merge Sort Test\n");

int n, i;

System.out.println("Enter number of integer elements:-");

n = scan.nextInt();

int arr[] = new int[ n ];

System.out.println("\nEnter "+ n +" integer elements:-");

for (i = 0; i < n; i++)

arr[i] = scan.nextInt();

sort(arr, 0, n);

System.out.println("\nElements after sorting:-");

for (i = 0; i < n; i++)

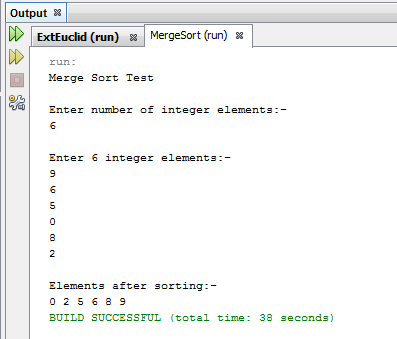
System.out.print(arr[i]+" ");

System.out.println();

}

}

Output:



LCS

Input:

package lcs;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class LCS {

int count=0;

public String lcs(String str1, String str2){

int l1 = str1.length();

int l2 = str2.length();

int[][] arr = new int[l1 + 1][l2 + 1];

for (int i = l1 - 1; i >= 0; i--){

for (int j = l2 - 1; j >= 0; j--){

if (str1.charAt(i) == str2.charAt(j))

arr[i][j] = arr[i + 1][j + 1] + 1;

else

arr[i][j] = Math.max(arr[i + 1][j], arr[i][j + 1]);

}

}

int i = 0, j = 0;

StringBuffer sb = new StringBuffer();

while (i < l1 && j < l2){

if (str1.charAt(i) == str2.charAt(j)){

sb.append(str1.charAt(i));

i++;

j++;

count++;

}

else if (arr[i + 1][j] >= arr[i][j + 1])

i++;

else

j++;

}

return sb.toString();

}

public static void main(String[] args) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Longest Common Subsequence Algorithm Test\n");

System.out.println("\nEnter string 1");

String str1 = br.readLine();

System.out.println("\nEnter string 2");

String str2 = br.readLine();

LCS obj = new LCS();

String result = obj.lcs(str1, str2);

System.out.println("\nLongest Common Subsequence : "+ result);

System.out.println("\nLength Of Longest Common Subsequence : "+

obj.count);

}

}

Output:

