****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)**

**LAB REPORT NO: 04**

**Course Title: Algorithms Lab**

**Course Code: CSE-206 Section: DC**

**Lab Experiment Name: LCS, LIS and Coin Change Problem in Java.**

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
|  | **Md. Sohanur Rahman** | **213902106** |

**Lab Date : 01.06.2023**

**Submission Date : 10.06.2023**

**Course Teacher’s Name : Md. Sultanul Islam Ovi**

**[For Teachers use only: Don’t Write Anything inside this box]**

|  |
| --- |
| **Lab Report Status**  **Marks: ………………………………… Signature: ...................**  **Comments: .............................................. Date: ..............................** |

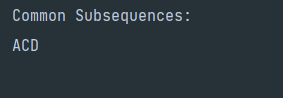
**\*\*\*\* Point to be noted: Explanation is given inside the code. \*\*\*\*\*\*\***

**1. Problem Statement:** Print all the common subsequence according to the descending order of the lengths for two given sequence.

**Code:**

*package* Algorithms\_Lab.LabReportFour;  
*//Print all common subsequence according to the descending order of the  
//lengths for two given sequence.  
  
import* java.util.ArrayList;  
*import* java.util.Collections;  
*import* java.util.*List*;  
  
*public class* CommonSubsequence {  
  
 *//Here we get common subsequence.  
 public static List*<String> getCommonSubsequences(String sequence1, String sequence2) {  
 *int*[][] dp = *new int*[sequence1.length() + 1][sequence2.length() + 1];  
  
 *for* (*int* i = 1; i <= sequence1.length(); i++) {  
 *for* (*int* j = 1; j <= sequence2.length(); j++) {  
 *if* (sequence1.charAt(i - 1) == sequence2.charAt(j - 1)) {  
 dp[i][j] = dp[i - 1][j - 1] + 1;  
 } *else* {  
 dp[i][j] = Math.*max*(dp[i - 1][j], dp[i][j - 1]);  
 }  
 }  
 }  
 *//Create a list of subsequence  
 List*<String> subsequences = *new* ArrayList<>();  
 *generateSubsequences*(dp, sequence1, sequence2, sequence1.length(), sequence2.length(), "", subsequences);  
 Collections.*sort*(subsequences, (s1, s2) -> s2.length() - s1.length());  
  
 *return* subsequences;  
 }  
  
 *//This method generate the subsequence.  
 private static void* generateSubsequences(*int*[][] dp, String sequence1, String sequence2, *int* i, *int* j,  
 String subsequence, *List*<String> subsequences) {  
 *if* (i == 0 || j == 0) {  
 subsequences.add(subsequence);  
 *return*;  
 }  
  
 *if* (sequence1.charAt(i - 1) == sequence2.charAt(j - 1)) {  
 *generateSubsequences*(dp, sequence1, sequence2, i - 1, j - 1, sequence1.charAt(i - 1) + subsequence, subsequences);  
 } *else* {  
 *if* (dp[i - 1][j] >= dp[i][j - 1]) {  
 *generateSubsequences*(dp, sequence1, sequence2, i - 1, j, subsequence, subsequences);  
 }  
 *if* (dp[i][j - 1] >= dp[i - 1][j]) {  
 *generateSubsequences*(dp, sequence1, sequence2, i, j - 1, subsequence, subsequences);  
 }  
 }  
 }  
  
 *public static void* main(String[] args) {  
 *//Declare two string.* String sequence1 = "ABCD";  
 String sequence2 = "ACDF";  
  
 *List*<String> commonSubsequences = *getCommonSubsequences*(sequence1, sequence2);  
  
  
 *// Finally Print the common subsequence.* System.***out***.println("Common Subsequences are:");  
 *for* (String subsequence : commonSubsequences) {  
 System.***out***.println(subsequence);  
 }  
 }  
}

**Output:**

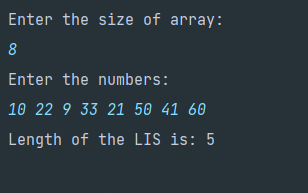


**2. Problem Statement:** Implement longest increasing subsequence problem using DP technique.

**Code:**

*package* Algorithms\_Lab.LabReportFour;  
*//Implement longest increasing subsequence problem using DP technique.  
  
import* java.util.Scanner;  
  
*public class* LCS {  
 *public static int* lis(*int*[] numbers){  
 *int*[] dp=*new int*[numbers.length];  
 *int* maxLength=1;  
 *//Initialize dp array with 1 as the minimum length of any  
 //subsequence is 1.  
 for*(*int* i=0;i< numbers.length;i++){  
 dp[i]=1;  
 }  
 *//Compute lis for each index.  
 for*(*int* i=1;i<numbers.length;i++){  
 *for*(*int* j=0;j<i;j++){  
 *if*(numbers[i]>numbers[j]){  
 dp[i]=Math.*max*(dp[i],dp[j]+1);  
 maxLength=Math.*max*(maxLength,dp[i]);  
 }  
 }  
 }  
 *return* maxLength;  
 }  
 *public static void* main(String[] args) {  
 Scanner scan=*new* Scanner(System.***in***);  
 System.***out***.println("Enter the size of array: ");  
 *int* size=scan.nextInt();  
 *int*[] numbers=*new int*[size];  
 System.***out***.println("Enter the numbers: ");  
 *for*(*int* i=0;i<size;i++){  
 numbers[i]=scan.nextInt();  
 }  
 *int* lisLength=*lis*(numbers);  
 *//Finally print the lis length.* System.***out***.println("Length of the LIS is: "+lisLength);  
 }  
}

**Output:**

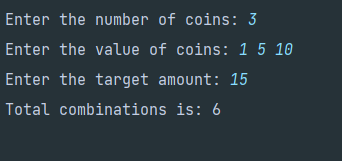
****

**3. Problem Statement:** Given a list of coins 1 taka, 5 taka and 10 taka, can you determine the total number of combinations of the coins in the given list to make up the number N taka.

**Code:**

*package* Algorithms\_Lab.LabReportFour;  
*//Task-3: Given a list of coins 1 taka, 5 taka and 10 taka, can you determine the total  
//number of combinations of the coins in the given list to make up the number N taka.  
  
import* java.util.Scanner;  
  
*public class* CoinCombinations {  
 *//This method calculated the coin combination.  
 //And return the combination at the end.  
 public static int* coinCombinations(*int*[] coins,*int* targetAmount){  
 *int*[] dp=*new int*[targetAmount+1];  
 dp[0]=1;  
 *for*(*int* coin : coins){  
 *for*(*int* i=coin;i<=targetAmount;i++){  
 dp[i]=dp[i]+dp[i-coin];  
 }  
 }  
 *return* dp[targetAmount];  
 }  
  
 *public static void* main(String[] args) {  
 Scanner scan=*new* Scanner(System.***in***);  
 *//Take input from user.* System.***out***.print("Enter the number of coins: ");  
 *int* n=scan.nextInt();  
 *int*[] coins=*new int*[n];  
 System.***out***.print("Enter the value of coins: ");  
 *for*(*int* i=0;i<n;i++){  
 coins[i]=scan.nextInt();  
 }  
 System.***out***.print("Enter the target amount: ");  
 *int* targetAmount=scan.nextInt();  
 *int* totalCombinations=*coinCombinations*(coins,targetAmount);  
 *//Finally print the total combination.* System.***out***.println("Total combinations is: "+totalCombinations);  
 }  
}

**Output:**

****