import java.util.\*;

public class LongestIncreasingSubsequence {

public static void main(String[] args) {

int[] arr = {10, 22, 9, 33, 21, 50, 41, 60};

int[] lis = *findLongestIncreasingSubsequence*(arr);

System.***out***.println("Length of longest increasing subsequence: " + lis.length);

System.***out***.print("Longest increasing subsequence: ");

for (int num : lis) {

System.***out***.print(num + " ");

}

}

public static int[] findLongestIncreasingSubsequence(int[] arr) {

if (arr == null || arr.length == 0) {

return new int[0];

}

int n = arr.length;

int[] dp = new int[n];

int[] prev = new int[n];

Arrays.*fill*(prev, -1);

dp[0] = 1;

int maxLen = 1;

int lastIndex = 0;

for (int i = 1; i < n; i++) {

dp[i] = 1;

for (int j = 0; j < i; j++) {

if (arr[i] > arr[j] && dp[j] + 1 > dp[i]) {

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

prev[i] = j;

}

}

if (dp[i] > maxLen) {

maxLen = dp[i];

lastIndex = i;

}

}

int[] lis = new int[maxLen];

int index = maxLen - 1;

while (lastIndex != -1) {

lis[index--] = arr[lastIndex];

lastIndex = prev[lastIndex];

}

return lis;

}

}