**Source Code**

**package** nonassisted;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**public** **class** LongestIncreasingSubsequence {

**public** **static** **void** main(String[] args) {

**int** array[] = {10, 22, 9, 33, 21, 50, 41, 60, 80};

ArrayList list = **new** ArrayList();

ArrayList longestList = **new** ArrayList();

**int** currentMax;

**int** highestCount = 0;

**for**(**int** i = 0; i < array.length;i++)

{

currentMax =0;

**for**(**int** j = i;j < array.length; j++)

{

**if**(array[j] > currentMax)

{

list.add(array[j]);

currentMax = array[j];

}

}

//Compare previous highest subsequence

**if**(highestCount < list.size())

{

highestCount = list.size();

longestList = **new** ArrayList(list);

}

list.clear();

}

System.***out***.println();

//Print list

Iterator itr = longestList.iterator();

System.***out***.println("The Longest subsequence");

**while**(itr.hasNext())

{

System.***out***.print(itr.next() + " ");

}

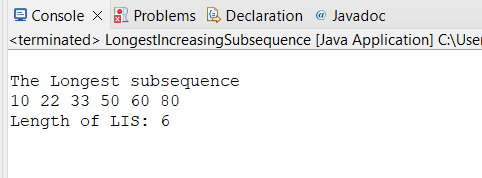
System.***out***.println();

System.***out***.println("Length of LIS: " + highestCount);

}

}

**Solution**

****