**Longest Increasing SubSequence Source Code**

package com.lis;

import java.util.Scanner;

public class LongIncSeq {

static int lis(int arr[], int n)

{

int lis[] = new int[n];

int i, j, max = 0;

/\* Initialize LIS values for all indexes \*/

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

lis[i] = 1;

/\* Compute optimized LIS values in bottom up manner \*/

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

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

if (arr[i] > arr[j] && lis[i] < lis[j] + 1)

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

/\* Pick maximum of all LIS values \*/

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

if (max < lis[i])

max = lis[i];

return max;

}

public static void main(String args[]) {

int n;

Scanner sc=new Scanner(System.in);

System.out.print("Enter the number of elements you want to store: ");

*\*reading the number of elements from the that we want to enter \**

n=sc.nextInt();

//creates an array in the memory of length 100

int[] arr = new int[100];

System.out.println("Enter the elements of the array: ");

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

{

*\*reading array elements from the user \**

arr[i]=sc.nextInt();

}

System.out.println("Longest Increasing subsequence : ");

System.out.print(lis(arr, n));

}

}