**package** LongestSubsequence;

**public** **class** LIS {

**static** **int** *max\_ref*;

**static** **int** \_lis(**int** arr[], **int** n)

{

// base case

**if** (n == 1)

**return** 1;

**int** res, max\_ending\_here = 1;

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

res = *\_lis*(arr, i);

**if** (arr[i - 1] < arr[n - 1]

&& res + 1 > max\_ending\_here)

max\_ending\_here = res + 1;

}

**if** (*max\_ref* < max\_ending\_here)

*max\_ref* = max\_ending\_here;

**return** max\_ending\_here;

}

// The wrapper function for \_lis()

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

{

*max\_ref* = 1;

*\_lis*(arr, n);

// returns max

**return** *max\_ref*;

}

// driver program to test above functions

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

{

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

**int** n = arr.length;

System.***out***.println("Length of lis is " + *lis*(arr, n)

+ "\n");

}

}