**package** p3;

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

**public** **static** **int** lis(**int**[] nums) {

**if** (nums == **null** || nums.length == 0) {

**return** 0;

}

**int** n = nums.length;

**int**[] dp = **new** **int**[n];

**int** maxLength = 1;

// Initialize the dp array with 1 as the minimum LIS length for each element

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

dp[i] = 1;

}

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

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

**if** (nums[i] > nums[j] && dp[i] < dp[j] + 1) {

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

maxLength = Math.*max*(maxLength, dp[i]);

}

}

}

**return** maxLength;

}

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

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

**int** longestIncreasingSubsequenceLength = *lis*(nums);

System.***out***.println("Length of the Longest Increasing Subsequence: " + longestIncreasingSubsequenceLength);

}

}