#include <stdio.h>

int longestNonDecreasingSubsequence(int arr[], int n) {

int dp[n]; // Array to store the length of LNDS at each position

int i, j, maxLength = 1;

// Initialize dp array: every element is an LNDS of length 1 by itself

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

dp[i] = 1;

}

// Fill dp array using the recurrence relation

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

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

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

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

}

}

}

// Find the maximum value in dp[] which will be the length of the LNDS

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

if (dp[i] > maxLength) {

maxLength = dp[i];

}

}

return maxLength;

}

int main() {

//int arr[] = {3, 4, 2, 8, 10, 5, 1}; // Example array

//int n = sizeof(arr) / sizeof(arr[0]);

int n;

scanf("%d",&n);

int arr[n];

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

scanf("%d",&arr[i]);

int result = longestNonDecreasingSubsequence(arr, n);

printf("%d\n", result);

return 0;

}

