//longest increasing subsequence - dynamic programming

#include<iostream>

using namespace std;

int max(int \*a,int n)

{

int maxm=a[0];

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

{

if(maxm<a[i])

maxm=a[i];

}

return maxm;

}

int longestIncreasingSubsequence(int x[],int n){

int\* t=(int\*)malloc(sizeof(int)\*n) ;

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

t[i] = 1;

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

{

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

{

if(x[i]>x[j] && t[i] < t[j] + 1)

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

}

}

free(t);

return max(t,n);

}

int main()

{

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

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

cout<<longestIncreasingSubsequence(arr,n);

return 0;

}