Longest Subsequence

Find the length of the longest (adjacent) increasing subsequence in a vector with n elements. As an example, consider the following vector that contains 12 elements.

5	8	10	5	2	1	12	12	83	30	40	65

The elements that form increasing sequences (subsequences of the entire array) in this array are.

5, 8, 10 (number of elements is 3)

5 (1)

2 (1)

1, 12, 12, 83 (4)

30, 40, 65 (3)

Therefore, the length of the longest increasing subsequence is 4. Notice that equal values are counted as increasing values.

Practice Problem

Find the largest value in a vector with n elements. Assume that n >= 1.

Base Name: increasing_sequences

Function prototypes

int increasing_sequences_iterative(std::vector<int> &numbers);

int increasing_sequences_recursive(std::vector<int> &numbers, int startIdx);

Both function prototypes get stored in project2.hpp.

In either of the two, numbers.size() will produce the number of elements in "numbers". In the recursive version, startIdx is initially 0. However, in the subsequent calls, it approaches numbers.size().

Hint

We are not looking for an optimal solution to this problem.

Output

A single integer -- the length of the longest increasing subsequence of the input vector.