AlgoExpert **Quad Layout** Sublime Monokai 00:00:00 **12px** 

Prompt Run Code

Solution 1

49 } 50

```
Scratchpad
  Our Solution(s)
                    Video Explanation
```

 $1\,$  // Copyright @ 2020 AlgoExpert, LLC. All rights reserved. 3 #include <vector> 4 using namespace std;  $\begin{tabular}{ll} 6 & vector < int >> & getKnapsackItems (vector < vector < int >> & knapsackValues, \\ \end{tabular}$ vector<vector<int>> items, int weight); 9 // O(nc) time | O(nc) space 10 vector<vector<int>> knapsackProblem(vector<vector<int>> items, int capacity) { vector<vector<int>> knapsackValues(items.size() + 1, 11 12 vector<int>(capacity + 1, 0)); 13 for (int i = 1; i < items.size() + 1; i++) {</pre> int currentWeight = items[i - 1][1]; 14 15 int currentValue = items[i - 1][0]; for (int c = 0; c < capacity + 1; c++) {</pre> 16 17 if (currentWeight > c) { knapsackValues[i][c] = knapsackValues[i - 1][c]; 18 19 } else { 20 knapsackValues[i][c] = max(knapsackValues[i - 1][c], 21 22 23 24 25 26 return getKnapsackItems(knapsackValues, items, 27 knapsackValues[items.size()][capacity]); 28 29 30 vector<vector<int>> getKnapsackItems(vector<vector<int>> knapsackValues, 31 vector<vector<int>> items, int weight) { vector<vector<int>> solution = {{}}, {}}; 32 int i = knapsackValues.size() - 1; 33 34 int c = knapsackValues[0].size() - 1; 35 while (i > 0) {  $\textbf{if} \ (\texttt{knapsackValues}[\texttt{i}][\texttt{c}] == \texttt{knapsackValues}[\texttt{i} - \textbf{1}][\texttt{c}]) \ \{ \\$ 36 37 i--; } else { 38 solution[1].insert(solution[1].begin(), i - 1); 39 c -= items[i - 1][1]; 40 41 i--; 42 43 **if** (c == 0) { 44 break; 45 46 47 solution[0].push\_back(weight); return solution; 48