```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace Algorithms
7  {
8      class Knapsack
9      {
10         double bestValue;
11         bool[] bestItems;
12         double[] itemValues;
13         double[] itemWeights;
14         double weightLimit;
15
16         void SolveRecursive(bool[] chosen, int depth, double currentWeight, double
               currentValue, double remainingValue)
17         {
18             if (currentWeight > weightLimit) return;
19             if (currentValue + remainingValue < bestValue) return;
20             if (depth == chosen.Length)
21             {
22                 bestValue = currentValue;
23                 System.Array.Copy(chosen, bestItems, chosen.Length);
24                 return;
25             }
26             remainingValue -= itemValues[depth];
27             chosen[depth] = false;
28             SolveRecursive(chosen, depth + 1, currentWeight, currentValue,
                 remainingValue);
29             chosen[depth] = true;
30             currentWeight += itemWeights[depth];
31             currentValue += itemValues[depth];
32             SolveRecursive(chosen, depth + 1, currentWeight, currentValue,
                 remainingValue);
33         }
34
35         public bool[] Solve()
36         {
37             var chosen = new bool[itemWeights.Length];
38             bestItems = new bool[itemWeights.Length];
39             bestValue = 0.0;
40             double totalValue = 0.0;
41             foreach (var v in itemValues) totalValue += v;
42             SolveRecursive(chosen, 0, 0.0, 0.0, totalValue);
43             return bestItems;
44         }
45     }
46 }
47
```