```
//Knapsack Problemi (Çanta Doldurma)
using System;
using System.Collections;
using System.Collections.Generic;

namespace Dinamic1
{
 class Program
 {
  public class Jewel
  {
   public int Weight { get; set; }
   public int Value { get; set; }
  }
  public static int KnapRec(int n, int c, List<Jewel> jewels)
  {
   int result = 0;

   if (n == -1 || c == -1)
   {
    result = 0;
   }
   else if (jewels[n].Weight > c)
   {
    result = KnapRec(n - 1, c, jewels);
   }
   else
   {
    int tmp1 = KnapRec(n - 1, c, jewels);
    int tmp2 = jewels[n].Value + KnapRec(n - 1, c - jewels[n].Weight, jewels);
    result = Math.Max(tmp1, tmp2);
   }
   return result;
  }

  public static int Knapsack(int bagCapacity, List<Jewel> jewels)
  {
   var itemCount = jewels.Count;

   int[,] matrix = new int[itemCount + 1, bagCapacity + 1];

   //Her öğeyi gözden geçirin.
   for (int i = 0; i <= itemCount; i++)
   {
    //Bu döngü temelde 0'da başlar ve yavaş yavaş büyür.
    //Daha küçük çantalara sığdırmanın en iyi yolunu bulmak gibi düşünün ve
    //ardından bunun üzerine inşa etmeye devam edin.
    for (int w = 0; w <= bagCapacity; w++)
    {
     //İlk döngüdeysek, başlangıç matris değerimizi 0 olarak ayarlayın.
     if (i == 0 || w == 0)
     {
      matrix[i, w] = 0;
      continue;
     }
```

```
    //Dizinler 0'dan başladığı için,
    //Bunu burada yaparsak okumak daha kolay, bu yüzden "önceki" öğeyi vb. okuduğumuzu
düşünmüyoruz.
    var currentJewelIndex = i - 1;
    var currentJewel = jewels[currentJewelIndex];
    //Mevcut mücevherin ağırlığı W'dan az mı
    //(Örneğin, mecbur kalırsak, başka bir şeyi boşaltsak bile çantaya koyabileceğimiz bir yer bulabilir
miyiz?)
    if (currentJewel.Weight <= w)
    {
     // Bu mücevheri hemen alıp diğer mücevherlerle birleştirsem
     // Bu, şu anda en iyi çaba olduğunu düşündüğünüzden daha büyük olur mu?
     // Başka bir deyişle, W 50 ise ve 30 ağırlığım varsa. 20 olan başka bir mücevheri birleştirirsem (veya
20 ağırlığında birden fazla veya hiç)
     // Bu kombinasyonla şu anda sahip olduğunuzdan daha mı iyi olurum?
     // Değilse, değeri son öğeyle ne olmuşsa öyle ayarlayın
     // (uygun olabilir, aynı şeyi yapmış olabilir ve uygun olmayabilir ve öncekini almış olabilir vb.).
     matrix[i, w] = Math.Max(currentJewel.Value + matrix[i - 1, w - currentJewel.Weight]
          , matrix[i - 1, w]);
    }
    // Bu mücevher sığamaz, bu yüzden son değerin ne olduğunu öne çıkarın
    // çünkü bu hala sahip olduğumuz "en iyi" uyum.
    else
     matrix[i, w] = matrix[i - 1, w];
   }
  }

  // Her şeyi ileri taşıdığımız için, her iki dizindeki en son öğe maksimum değerimizdir
  return matrix[itemCount, bagCapacity];
 }
 static void Main(string[] args)
 {
  var items = new List<Jewel>
 {
  new Jewel {Value = 120, Weight = 10},
  new Jewel {Value = 100, Weight = 20},
  new Jewel {Value = 500, Weight = 30},
  new Jewel {Value = 400, Weight = 25},
  new Jewel {Value = 200, Weight = 5}
 };

  //Console.WriteLine(Knapsack(50, items));
  //Console.WriteLine(KnapRec(4,50, items));
  Console.ReadKey();
 }
}


}
```