# Feedback — Programming Assignment #3  

You submitted this quiz on **Mon 20 Apr 2015 11:33 AM IST**. You got a score of **5.00** out of **5.00**.

## Question 1

In this programming problem and the next you'll code up the knapsack algorithm from lecture. Let's start with a warm-up. Download the text file here. This file describes a knapsack instance, and it has the following format:

[knapsack_size][number_of_items]

[value_1] [weight_1]

[value_2] [weight_2]

...

For example, the third line of the file is "50074 659", indicating that the second item has value 50074 and size 659, respectively.

You can assume that all numbers are positive. You should assume that item weights and the knapsack capacity are integers.

In the box below, type in the value of the optimal solution.

ADVICE: If you're not getting the correct answer, try debugging your algorithm using some small test cases. And then post them to the discussion forum!

**You entered:**

| 2493893 |
|---|

| Your Answer | | Score | Explanation |
|---|---|---|---|
| 2493893 | ✔ | 1.00 | |
| Total | | 1.00 / 1.00 | |

## Question 2

This problem also asks you to solve a knapsack instance, but a much bigger one.

Download the text file here. This file describes a knapsack instance, and it has the following format:

[knapsack_size][number_of_items]

[value_1] [weight_1]

[value_2] [weight_2]

...

For example, the third line of the file is "50074 834558", indicating that the second item has value 50074 and size 834558, respectively. As before, you should assume that item weights and the knapsack capacity are integers.

This instance is so big that the straightforward iterative implemetation uses an infeasible amount of time and space. So you will have to be creative to compute an optimal solution. One idea is to go back to a recursive implementation, solving subproblems --- and, of course, caching the results to avoid redundant work --- only on an "as needed" basis. Also, be sure to think about appropriate data structures for storing and looking up solutions to subproblems.

In the box below, type in the value of the optimal solution.

ADVICE: If you're not getting the correct answer, try debugging your algorithm using some small test cases. And then post them to the discussion forum!

**You entered:**

4243395

| Your Answer | | Score | Explanation |
|---|---|---|---|
| 4243395 | ✔ | 4.00 | |
| Total | | 4.00 / 4.00 | |