CSC 364 Assignment #2
Total Points: 40

Knapsack Problem

The file KnapsackData1.txt and KnapsackData2.txt, provided on Canvas. These files are sample input files for the following Knapsack Problem that you will solve.

KnapsackData1.txt contains a list of four prospective projects for the upcoming year for a particular company:

```
Project0 6 30
Project1 3 14
Project2 4 16
Project3 2 9
```

Each line in the file provides three pieces of information:

1) String: The name of the project;
2) Integer: The amount of employee labor that will be demanded by the project, measured in work weeks;
3) Integer: The net profit that the company can expect from engaging in the project, measured in thousands of dollars.

Your task is to write a program that:

1) Prompts the user for the number of work weeks available (integer);
2) Prompts the user for the name of the input file (string);
3) Prompts the user for the name of the output file (string);
4) Reads the available projects from the input file;
5) Solves the corresponding knapsack problem, without repetition of items; and
6) Writes to the output file a summary of the results, including the expected profit and a list of the best projects for the company to undertake.

Here is a sample session with the program:

```
Enter the number of available employee work weeks: 10
Enter the name of input file: KnapsackData1.txt
Enter the name of output file: Output1.txt
Number of projects = 4
Done
```

For the above example, here is the output that should be written to Output1.txt:

```
Number of projects available: 4
Available employee work weeks: 10
Number of projects chosen: 2
Total profit: 46
Project0 6 30
Project2 4 16
```

The file KnapsackData2.txt, also provided on Canvas, contains one thousand prospective projects. Your program should also be able to handle this larger problem as well. The corresponding output file, Output2.txt, is also on Canvas.

With a thousand prospective projects to consider, it will be impossible for your program to finish in a reasonable amount of time if it uses a "brute-force search" that explicitly considers every possible combination of projects. You are required to use a dynamic programming approach to this problem, as discussed in class.

Submission:

Submit your source code file on Canvas: ***Knapsack.java***. Be sure to follow the coding conventions described below. Make sure you follow the coding conventions mentioned in CodingConventions.pdf on Canvas.