Assignment #3

A.) We are given a number of objects and a knapsack. The knapsack has a weight capacity of W and each object has an associated value v and weight w. Our goal is collect the maximum value in our knapsack while respecting the weight constraint. We are allowed to either take an object or leave it behind, no fractional objects are allowed. Our goal may be stated as follows:

$$\text{maximize} \sum_{i=1}^{n} x_i v_i \text{ subject to } \sum_{i=1}^{n} x_i w_i \leq W$$

where $v_i > 0$, $w_i > 0$, and $x_i \in \{0, 1\}$ for $1 \leq i \leq n$.

Devise a dynamic programming algorithm to solve this problem.

Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your write-up should provide the following:

1. A description of the algorithm in English and pseudocode.
2. At least one worked example to show how your algorithm works.
3. A proof (or argument) the algorithm is correct.
4. An analysis of the running time of the algorithm.

B.) Implement your algorithm in C++. You may use the C++ STL. The input to your program is a file formatted as follows. The first line contains two space separated integers, W and the number of available objects n. Each of the n subsequent lines contains the value of the i^{th} object followed by its weight. All integer values are space separated.

For example, if the knapsack capacity is 10 and we have 4 objects with values 7, 2, 8, and 11 with weights 2, 3, 2, and 12. The file looks like this:

```
10 4
7 2
2 3
8 2
11 12
```

Due date:

Completed assignment - Wednesday 13th at 11:59 p.m. (one minute before midnight)

Program submission:

See the class web page for "How to submit programs".

Program evaluation:

See the "Program evaluation worksheet" for programming assignments on the course web page. Your program must meet all of these criteria in order to receive full credit.