Net ID: wpicc001
Name: William Patrick Picca

Sources Used:
Communicated with fellow student: Amneh Alsuqi

Problem A: Merge The Candies
Submission ID: 133305800

For the first programming assignment I started off by taking the first input for the total number of bags and then for the remaining input I created and stored the values into a vector. I choose to do a vector because it's much easier later on in the code to simply remove a given element from the vector than to use an array and keep track of what element positions we are no longer using.

I then choose to perform an insertion sort on the vector, and then afterwards create a temp vector for the purpose of flipping the vector, so it's then in descending order.

Afterwards I began the main focus of the assignment by using a while loop dependent upon the size of the vector. For every iteration through this loop, I add up the 2 lowest values within the vector, I proceed to sort it just as I have done before, and then remove the 2 numbers I just used to obtain the sum. I add this sum value to a counting variable to keep track, and then add this sum as a new element into my vector. This process repeats until only 1 number remains in the vector, at which point all the different bags have been combined into one.

At the end I double the candies and this is our final result for any inputted set of numbers.

Problem B: The simplest knapsack problem in the world
Submission ID: 133302051

For the second programming assignment it starts quite simple. Take in the input for the weight budget and given number of items. Then take in and store the list of items and their respective values into arrays.

At which point it then enters the recursively calling knapsack function, iterating through all possible weights and values through a nested for loop, checking if any given item is within the weight limit. Then checks if that item, if added to our final solution, would result in a higher value solution.

We would store the optimal answer into a temporary vector, and after iterating through all possible items and values, we would return the final combination that would be our highest possible value.