**Fractional Knapsack**

Given *weights* and *values* of **N** items, we need to put these items in a knapsack of capacity **W** to get the *maximum* total value in the knapsack.  
**Note:** Unlike 0/1 knapsack, you are allowed to break the item.

**Example 1:**

**Input:**

N = 3, W = 50

values[] = {60,100,120}

weight[] = {10,20,30}

**Output:**

240.00

**Explanation:**Total maximum value of item

we can have is 240.00 from the given

capacity of sack.

**Example 2:**

**Input:**

N = 2, W = 50

values[] = {60,100}

weight[] = {10,20}

**Output:**

160.00

**Explanation:**

Total maximum value of item

we can have is 160.00 from the given

capacity of sack.

**Your Task** :  
Complete the function ***fractionalKnapsack****()* that receives maximum capacity , array of structure/class and size n and returns a double value representing the maximum value in knapsack.  
**Note:**The details of structure/class is defined in the comments above the given function.

**Expected Time Complexity** : O(NlogN)  
**Expected Auxilliary Space**: O(1)

**Constraints:**  
1 <= N <= 105  
1 <= W <= 105