Fractional Knapsack

Unlike 0/1 Knapsack we can take a fraction of weight to maximise the value. We will be given two arrays value and weight our motive is to maximise the value given a knapsack of W weight. Mtlb ki hme W weight ki knapsack bnani h lkn uski value maximise krke.

Approach is to think greedy agr hm pehle hi value nikal le per unit weight as sort krde aray ko in descending order then our work becomes easy hme try krenge ki jyada se jyada weight include krpaye of greater value. Agr W less h current arr[i]k weight se so we willinclude the whole value lkn agr bda h to hme fraction lega uske liye hm 1unit ki value nikalenge and jitna bhi weight aurbcha hua h usse multiply krke value me add krdenge then we will return the sum of all values as ans.

```
bool comparator(Item a,Item b)
{
    double a1=(double)a.value/a.weight;
    double b1=(double)b.value/b.weight;
    return a1>b1;
}
class Solution
{
    public:
    //Function to get the maximum total value in the knapsack.
    double fractionalKnapsack(int W, Item arr[], int n)
    {
        // Your code here
        sort(arr,arr+n,comparator);
        double value_sum=0;
        for(int i=0;i<n;i++)
        {
            if(arr[i].weight<=W)
            {
                value_sum+=arr[i].value;
                W-=arr[i].weight;
            }
            else
            {
                double remain=(double)arr[i].value/arr[i].weight*W;
                value_sum+=remain;
                break;
            }
        }
        return value_sum;
    }
```

```
};
```