In [1]:

```
# Assignment 3:Write a program to solve a fractional Knapsack problem using a greedy meth
od
```

In [22]:

```
class ItemValue:
    def __init__(self, wt, val, ind):
        self.wt = wt
        self.val = val
        self.ind = ind
        self.cost = val // wt
    def __lt__(self, other):
        return self.cost < other.cost

    # Greedy Approach
class FractionalKnapSack:

    @staticmethod
    def getMaxValue(wt, val, capacity):
        iVal = []
        for i in range(len(wt)):
            iVal.append(ItemValue(wt[i], val[i], i))

           # sorting items by value
        iVal.sort(reverse=True)

        totalValue = 0
        for i in iVal:
            curWt = int(i.wt)
            curVal = int(i.val)
            if capacity - curWt >= 0:
                capacity -= curWt
                totalValue += curVal
            else:
                fraction = capacity / curWt
                totalValue += curVal * fraction
                capacity = int(capacity - (curWt * fraction))
                break
        return totalValue
 # Driver Code
if __name__ == "__main__":
    wt = [2,3,5,7,1,4,1]
    val = [10,5,15,7,6,18,3]
    capacity = 15

maxValue = FractionalKnapSack.getMaxValue(wt, val, capacity)
print("Maximum value in Knapsack =", maxValue)
```

```
Maximum value in Knapsack = 55.333333333333336
```

In [ ]:

```

```