

```

// Main greedy function to solve problem
double fractionalKnapsack(int W, struct Item arr[], int n)
{
    //sorting Item on basis of ratio
    sort(arr, arr + n, cmp);
    int curWeight = 0; // Current weight in knapsack
    double finalvalue = 0.0; // Result (value in Knapsack)

    // Looping through all Items
    for (int i = 0; i < n; i++)
    {
        // If adding Item won't overflow, add it completely
        if (curWeight + arr[i].weight <= W)
        {
            curWeight += arr[i].weight;
            finalvalue += arr[i].value;
        }

        // If we can't add current Item, add fractional part of it
        else
        {
            int remain = W - curWeight;
            finalvalue += arr[i].value * ((double)remain / (double)arr[i].weight);
            break;
        }
    }

    // Returning final value
    return finalvalue;
}

```