

```

#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

struct Item {
    int weight;
    int value;
    double ratio;
};

bool compareItems(const Item &item1, const Item &item2) {
    return item1.ratio > item2.ratio;
}

double fractionalKnapsack(vector<Item> items, int capacity) {
    int n = items.size();
    for (int i = 0; i < n; ++i) {
        items[i].ratio = static_cast<double>(items[i].value) /
items[i].weight;
    }
    sort(items.begin(), items.end(), compareItems);

    double totalValue = 0.0;
    int currentWeight = 0;

    for (int i = 0; i < n; ++i) {
        if (currentWeight + items[i].weight <= capacity) {
            currentWeight += items[i].weight;
            totalValue += items[i].value;
        } else {
            int remainingWeight = capacity - currentWeight;
            totalValue += (static_cast<double>(remainingWeight) /
items[i].weight) * items[i].value;

```

```

        break;
    }
}
return totalValue;
}

int main() {
    int n, capacity;
    cout << "Enter the number of items: ";
    cin >> n;

    vector<Item> items(n);
    cout << "Enter the weight and value of each item:" << endl;
    for (int i = 0; i < n; ++i) {
        cin >> items[i].weight >> items[i].value;
    }

    cout << "Enter the maximum capacity of the knapsack: ";
    cin >> capacity;
    double maxValue = fractionalKnapsack(items, capacity);
    cout << "Maximum value in the knapsack: " << maxValue << endl;
    return 0;
}

```

//OUTPUT:

```

Enter the number of items: 3
Enter the weight and value of each item:
10 60
20 100
30 120
Enter the maximum capacity of the knapsack: 50
Maximum value in the knapsack: 240

```