Exp_07\KnapsackGreedy.c

```c
#include <stdio.h>
#include <stdlib.h>

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

int compare(const void *a, const void *b) {
    Item *itemA = (Item *)a;
    Item *itemB = (Item *)b;
    if (itemB->ratio > itemA->ratio) return 1;
    if (itemB->ratio < itemA->ratio) return -1;
    return 0;
}

void knapsackGreedy(int weights[], int values[], int n, int capacity) {
    Item items[n];

    // Calculate value-to-weight ratio
    for (int i = 0; i < n; i++) {
        items[i].weight = weights[i];
        items[i].value = values[i];
        items[i].ratio = (double)values[i] / weights[i];
        items[i].index = i + 1;
    }

    // Sort by ratio in descending order
    qsort(items, n, sizeof(Item), compare);

    double totalValue = 0.0;
    int remainingCapacity = capacity;
    double fractions[n];

    for (int i = 0; i < n; i++) {
        fractions[i] = 0.0;
    }

    printf("\nItem\tWeight\tValue\tRatio\t\tFraction\n");
    printf("------------------------------------------------------------\n");

    for (int i = 0; i < n; i++) {
        if (remainingCapacity >= items[i].weight) {
            // Take full item
            fractions[items[i].index - 1] = 1.0;
            totalValue += items[i].value;
            remainingCapacity -= items[i].weight;
            printf("%d\t%d\t%d\t%.2f\t\t%.2f\n", items[i].index, items[i].weight,
                   items[i].value, items[i].ratio, 1.0);
        } else if (remainingCapacity > 0) {
            // Take fractional item
            double fraction = (double)remainingCapacity / items[i].weight;
            fractions[items[i].index - 1] = fraction;
            totalValue += items[i].value * fraction;
            printf("%d\t%d\t%d\t%.2f\t\t%.2f\n", items[i].index, items[i].weight,
                   items[i].value, items[i].ratio, fraction);
            remainingCapacity = 0;
            break;
        }
    }

    printf("------------------------------------------------------------\n");
    printf("Maximum value: %.2f\n", totalValue);
}

int main() {
    int n, capacity;

    printf("Enter number of items: ");
    scanf("%d", &n);

    int weights[n], values[n];

    printf("Enter weights: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &weights[i]);
    }

    printf("Enter values: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &values[i]);
    }

    printf("Enter knapsack capacity: ");
    scanf("%d", &capacity);

    knapsackGreedy(weights, values, n, capacity);

    return 0;
}
```