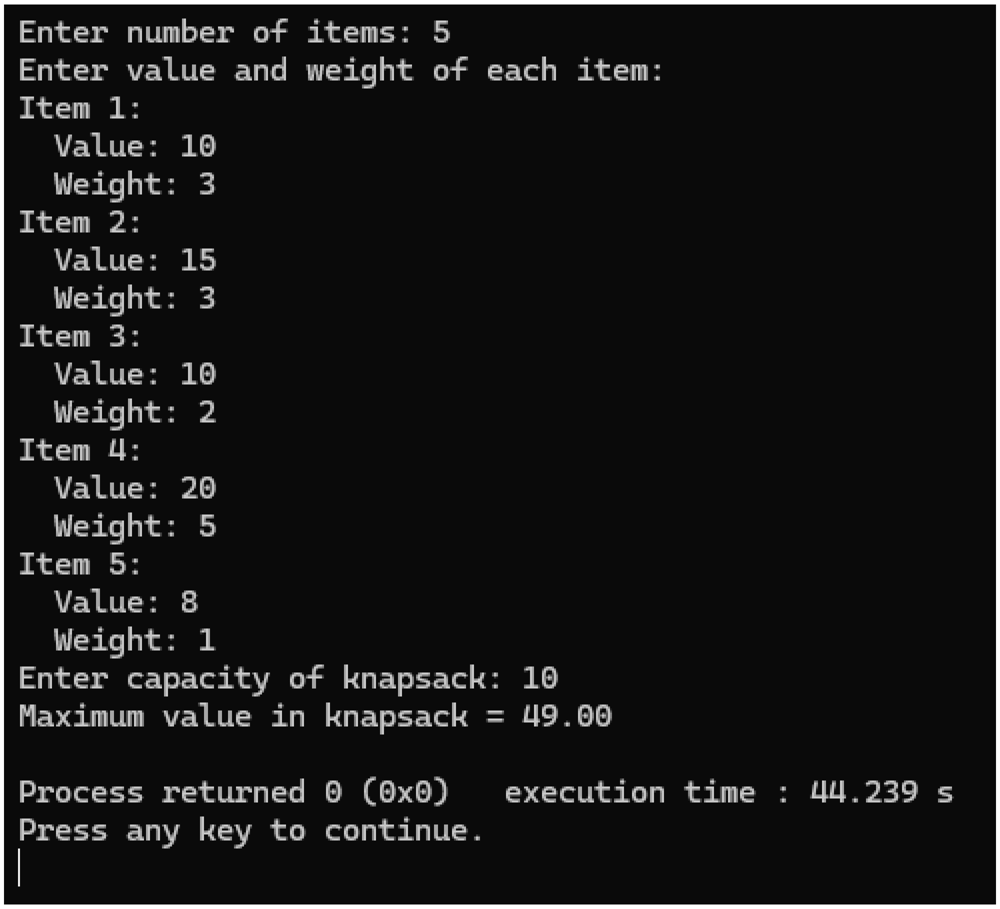
Implement fractional knapsack problem using Greedy technique.

Code:

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

// Structure to store weight, value and value/weight ratio
typedef struct {
    int value;
    int weight;
    float ratio;
} Item;

// Comparison function for sorting items by decreasing value/weight ratio
int compare(const void *a, const void *b) {
    Item *item1 = (Item *)a;
    Item *item2 = (Item *)b;
    if (item1->ratio < item2->ratio) return 1;
    else if (item1->ratio > item2->ratio) return -1;
    else return 0;
}

// Function to implement fractional knapsack
float fractionalKnapsack(Item items[], int n, int capacity) {
    qsort(items, n, sizeof(Item), compare);

    float totalValue = 0.0;
    int i;

    for (i = 0; i < n && capacity > 0; i++) {
        if (items[i].weight <= capacity) {
            totalValue += items[i].value;
            capacity -= items[i].weight;
        } else {
            totalValue += items[i].ratio * capacity;
            break;
        }
    }

    return totalValue;
}

int main() {
    int n, capacity;

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

    Item items[n];

    printf("Enter value and weight of each item:\n");
    for (int i = 0; i < n; i++) {
        printf("Item %d:\n", i + 1);
        printf("  Value: ");
        scanf("%d", &items[i].value);
        printf("  Weight: ");
        scanf("%d", &items[i].weight);
        items[i].ratio = (float)items[i].value / items[i].weight;
    }

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

    float maxValue = fractionalKnapsack(items, n, capacity);
    printf("Maximum value in knapsack = %.2f\n", maxValue);

    return 0;
}
```

Output:



```
Enter number of items: 5
Enter value and weight of each item:
Item 1:
  Value: 10
  Weight: 3
Item 2:
  Value: 15
  Weight: 3
Item 3:
  Value: 10
  Weight: 2
Item 4:
  Value: 20
  Weight: 5
Item 5:
  Value: 8
  Weight: 1
Enter capacity of knapsack: 10
Maximum value in knapsack = 49.00

Process returned 0 (0x0)   execution time : 44.239 s
Press any key to continue.
```