Aim: write a program in c to perform knapsack problem using greedy method

Algorithm for Knapsack problem using Greedy method:

1. Sort the items in descending order of their value-to-weight ratio.
2. Pick items in the sorted order, adding them to the knapsack until it is full.
3. Calculate the final value of the knapsack by adding up the values of the items picked.

Source Code:

#include <stdio.h>

#include <stdlib.h>

typedef *struct* {

*int* value;

*int* weight;

*float* ratio;

} Item;

*int* compare(const *void*\* a, const *void*\* b)

{

    Item \*ia = (Item \*)a;

    Item \*ib = (Item \*)b;

    return (*int*)(100.0f \* ia->ratio - 100.0f \* ib->ratio);

}

*void* knapsack(*int* n, *int* W, Item arr[])

{

*int* curWeight = 0;

*int* finalValue = 0;

    for (*int* i = 0; i < n; i++) {

        if (curWeight + arr[i].weight <= W) {

            finalValue += arr[i].value;

            curWeight += arr[i].weight;

        }

        else {

*int* remain = W - curWeight;

            finalValue += arr[i].value \* ((*float*) remain / arr[i].weight);

            break;

        }

    }

    printf("Maximum value we can obtain = %d\n", finalValue);

}

*int* main()

{

*int* W, n;

    printf("Enter the maximum weight capacity of the knapsack: ");

    scanf("%d", &W);

    printf("Enter the number of items: ");

    scanf("%d", &n);

    Item arr[n];

    for (*int* i = 0; i < n; i++) {

        printf("Enter the value and weight of item %d: ", i + 1);

        scanf("%d%d", &arr[i].value, &arr[i].weight);

        arr[i].ratio = (*float*) arr[i].value / arr[i].weight;

    }

    qsort(arr, n, sizeof(arr[0]), compare);

    knapsack(n, W, arr);

    return 0;

}

Output:

Enter the maximum weight capacity of the knapsack: 50

Enter the number of items: 3

Enter the value and weight of item 1: 60

10

Enter the value and weight of item 2: 100

20

Enter the value and weight of item 3: 120

30

Maximum value we can obtain = 220