30.Knapsack Problem

#include <stdio.h>

#include <stdlib.h>

struct Item

{

int index;

int weight;

int profit;

float ratio;

};

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

{

struct Item \*itemA = (struct Item \*)a;

struct Item \*itemB = (struct Item \*)b;

return (itemB->ratio - itemA->ratio);

}

void knapsack(int knapsackWeight, struct Item items[], int numItems)

{

qsort(items, numItems, sizeof(struct Item), compareItems);

int currentWeight = 0;

float totalProfit = 0.0;

for (int i = 0; i < numItems; i++)

{

if (currentWeight + items[i].weight <= knapsackWeight)

{

currentWeight += items[i].weight;

totalProfit += items[i].profit;

printf("Item %d (Weight: %d, Profit: %d) added to knapsack\n",

items[i].index, items[i].weight, items[i].profit);

}

else

{

float remainingWeight = knapsackWeight - currentWeight;

totalProfit += (remainingWeight / items[i].weight) \* items[i].profit;

printf("Fractional part of Item %d (Weight: %d, Profit: %d) added to knapsack\n",

items[i].index, items[i].weight, items[i].profit);

break;

}

}

printf("Total Profit: %.2f\n", totalProfit);

}

int main()

{

int knapsackWeight, numItems;

printf("Enter the knapsack weight: ");

scanf("%d", &knapsackWeight);

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

scanf("%d", &numItems);

struct Item items[numItems];

printf("Enter item details (Weight Profit):\n");

for (int i = 0; i < numItems; i++)

{

items[i].index = i + 1;

scanf("%d %d", &items[i].weight, &items[i].profit);

items[i].ratio = (float)items[i].profit / items[i].weight;

}

knapsack(knapsackWeight, items, numItems);

return 0;

}

OUTPUT

