Experiment No.5

To implement Fractional Knap Sack using Greedy Method.

#include <stdio.h>

struct Item {

int value;

int weight;

};

double ratio[n];

for (int i = 0; i < n; i++) {

ratio[i] = (double)items[i].value / items[i].weight;

}

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (ratio[j] < ratio[j + 1]) {

double tempRatio = ratio[j];

ratio[j] = ratio[j + 1];

ratio[j + 1] = tempRatio;

struct Item tempItem = items[j];

items[j] = items[j + 1];

items[j + 1] = tempItem;

}

}

}

double totalValue = 0.0;

int currentWeight = 0;

for (int i = 0; i < n; i++) {

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

totalValue += items[i].value;

currentWeight += items[i].weight;

} else {

int remainingWeight = capacity - currentWeight;

totalValue += (double)remainingWeight / items[i].weight \* items[i].value;

break;

}

}

return totalValue;

}

int main() {

struct Item items[] = {{60, 10}, {100, 20}, {120, 30}};

int n = sizeof(items) / sizeof(items[0]);

int capacity = 50;

double maxValue = fractionalKnapsack(capacity, items, n);

printf("Maximum value in the knapsack: %.2lf\n", maxValue);

return 0;

}

Output:-

Maximum value in the knapsack: 240.00