DESIGN AND ANALYSIS OF ALGORITHMS

LAB EXPERIMENT-08

Name: RIYA KUMARI
SAP ID:590016221

Batch:34

Course: B.Tech. CSE (Sem III) **Subject:** DAA
Submitted to: Virendra Kumar **GitHub Repository Link:** https://github.com/Riyakumari1314

Objective: Implement and analyze the Fractional Knapsack algorithm using the Greedy approach, which selects items to maximize total profit within a given weight capacity.

Problem Statement: You are given a set of n items, each with a profit (value) and a weight. Your task is to determine the maximum profit that can be obtained by putting items (or fractions of items) into a knapsack of capacity W.

(i) Fractional Knapsack allows you to take a fraction of an item if the entire item cannot fit.

(ii) The selection criterion is based on the profit/weight ratio.

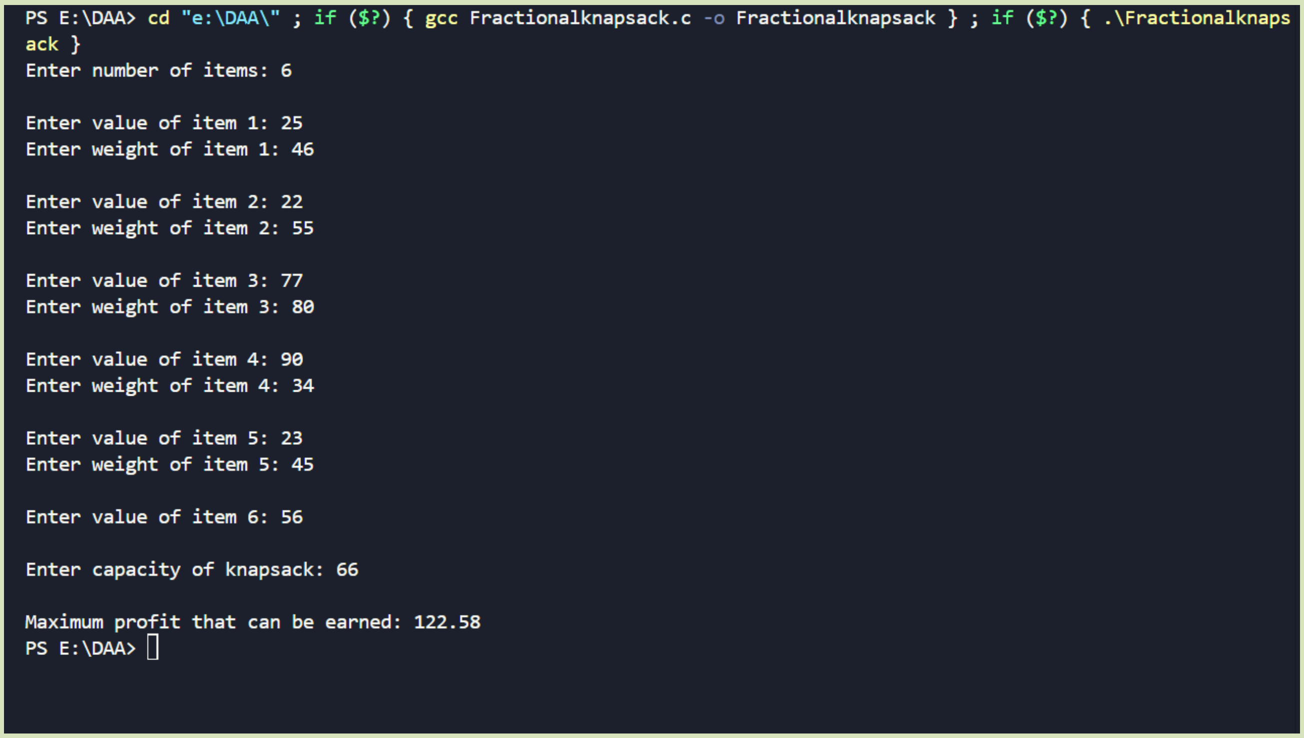
Input: Number of items n Profit (value) (pi) and weight (wi) of each item Capacity of the knapsack W

Output: Maximum profit

Code:

```
#include <stdio.h>

struct Item {
    int value;
    int wt;
    float density;
};
void arrangeByDensity(struct Item arr[], int size) {
    struct Item temp;
    for (int i = 0; i < size - 1; i++) {
        for (int j = i + 1; j < size; j++) {
            if (arr[i].density < arr[j].density) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int main() {
    int n;
    float capacity, profit = 0.0;
```

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

    struct Item arr[n];
    for (int i = 0; i < n; i++) {
        printf("\nEnter value of item %d: ", i + 1);
        scanf("%d", &arr[i].value);
        printf("Enter weight of item %d: ", i + 1);
        scanf("%d", &arr[i].wt);
        arr[i].density = (float)arr[i].value / arr[i].wt;
    }

    printf("\nEnter capacity of knapsack: ");
    scanf("%f", &capacity);

    arrangeByDensity(arr, n);
    for (int i = 0; i < n && capacity > 0; i++) {
        if (arr[i].wt <= capacity) {
            profit += arr[i].value;
            capacity -= arr[i].wt;
        } else {
            profit += arr[i].density * capacity;
            capacity = 0; // bag is full
        }
    }

    printf("\nMaximum profit that can be earned: %.2f\n", profit);

    return 0;
}
```


```
PS E:\DAA> cd "e:\DAA\" ; if ($?) { gcc Fractionalknapsack.c -o Fractionalknapsack } ; if ($?) { .\Fractionalknaps
ack }
Enter number of items: 6

Enter value of item 1: 25
Enter weight of item 1: 46

Enter value of item 2: 22
Enter weight of item 2: 55

Enter value of item 3: 77
Enter weight of item 3: 80

Enter value of item 4: 90
Enter weight of item 4: 34

Enter value of item 5: 23
Enter weight of item 5: 45

Enter value of item 6: 56

Enter capacity of knapsack: 66

Maximum profit that can be earned: 122.58
PS E:\DAA> 
```