

LAB TASK 7

1. Greedy Approach: Implementation of Fractional Knapsack.

Method 1: Minimum Weight

```
#include <stdio.h>

int main() {
    int n = 3;
    int value[3] = {60, 100, 120};
    int weight[3] = {10, 20, 30};
    int W = 50;

    for (int i = 0; i < n-1; i++) {
        for (int j = i+1; j < n; j++) {
            if (weight[i] > weight[j]) {
                int temp = weight[i]; weight[i] = weight[j]; weight[j] = temp;
                temp = value[i]; value[i] = value[j]; value[j] = temp;
            }
        }
    }

    double total = 0.0;
    for (int i = 0; i < n; i++) {
        if (weight[i] <= W) {
            W -= weight[i];
            total += value[i];
        } else {
            total += value[i] * ((double)W / weight[i]);
            break;
        }
    }

    printf("Max value (Min Weight First) = %.2f\n", total);
    return 0;
}
```

OUTPUT:

```
PS C:\Users\KIIT0001\DAA LAB> cd "c:\
in_weight_knapsack } ; if ($?) { .\mir
Max value (Min Weight First) = 240.00
PS C:\Users\KIIT0001\DAA LAB> █
```

Method 2:

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

int main()
{
    int n, w=50;
    printf("enter the number of items: \n");
    scanf("%d",&n);
    int profit[n], weight[n];
    printf("Enter the profit and weights of %d items: ",n);
    for(int i=0;i<n;i++)
    {
        scanf("%d %d",&profit[i],&weight[i]);
        printf("\n");
    }
    for (int i = 0; i < n-1; i++)
    {
        for (int j = i+1; j < n; j++)
        {
            if (profit[i] < profit[j])
            {
                int temp = profit[i]; profit[i] = profit[j]; profit[j] = temp;
                temp = weight[i]; weight[i] = weight[j]; weight[j] = temp;
            }
        }
    }
    double total = 0.0;
    for (int i = 0; i < n; i++) {
        if (weight[i] <= w) {
            w -= weight[i];
            total += profit[i];
        }
    }
}
```

```

} else {
    total += profit[i] * ((double)w / weight[i]);
    break;
}

}
printf("Total max profit: %f\n",total);
}

```

OUTPUT:

```

enter the number of items:
3
Enter the profit and weights of 3 items: 100 20

120 30

60 100

Total max profit: 220.000000
PS C:\Users\KIIT0001\DAA LAB>

```

Method 3: Taking Ratios

```

#include <stdio.h>

struct Item {
    int profit;
    int weight;
    float ratio;
};

void swap(struct Item *a, struct Item *b) {
    struct Item temp = *a;

```

```

    *a = *b;
    *b = temp;
}

int partition(struct Item arr[], int low, int high) {
    float pivot = arr[high].ratio;
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (arr[j].ratio > pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}

void quickSort(struct Item arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

float fractionalKnapsack(struct Item items[], int n, int capacity) {

    quickSort(items, 0, n - 1);

    float totalProfit = 0.0;
    for (int i = 0; i < n; i++) {
        if (items[i].weight <= capacity) {
            totalProfit += items[i].profit;
            capacity -= items[i].weight;
        } else {
            totalProfit += items[i].profit * ((float)capacity / items[i].weight);
            break;
        }
    }
}

```

```

    }
    return totalProfit;
}

int main() {
    int n, capacity;

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

    struct Item items[n];

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

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

    float maxProfit = fractionalKnapsack(items, n, capacity);
    printf("Maximum Profit: %.2f\n", maxProfit);

    return 0;
}

```

OUTPUT:

```

Enter number of items: 3
Enter profit and weight of items:
120 30
100 20
60 10
Enter capacity of knapsack: 50
Maximum Profit: 240.00
PS C:\Users\KIIT0001\DAA LAB>

```