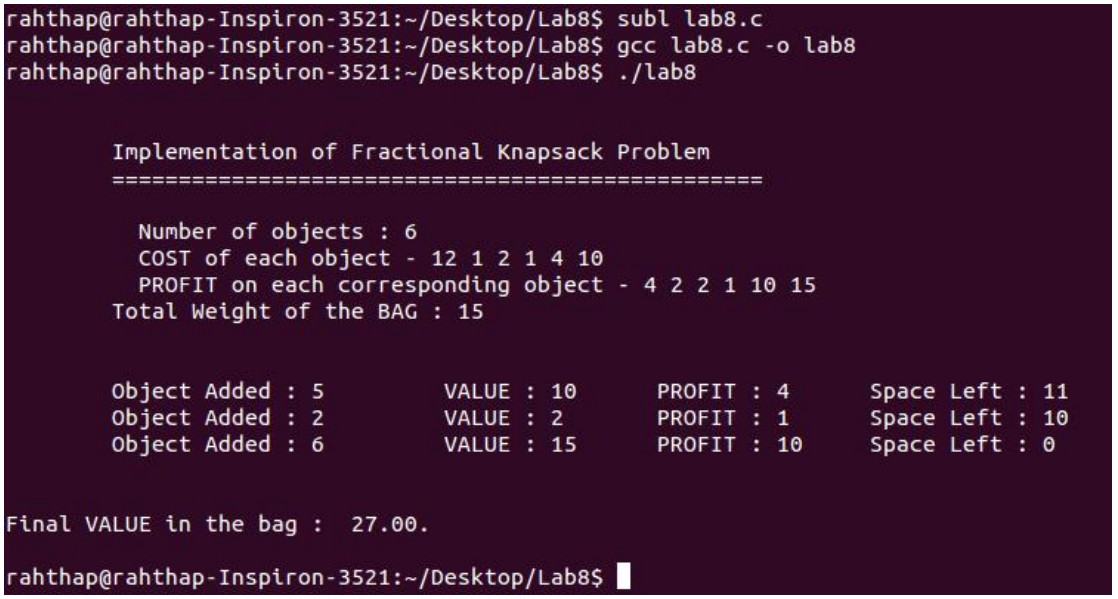
LAB 8

Fractional Knapsack Problem

Name : Rahul Thapar
ID : 1410110321
Date : 22 Feb, 2017

SCREENSHOT :



CODE:

/*

@author : Rahul Thapar
ID : 1410110321

Problem : Implement Fractional Knapsack Problem

Algorithm

1. Calculate DENSITY(PROFIT) : value per weight for each item
2. Sort the items as per the value density in descending

```
order
        3.  Take as much item as possible not already taken in the
knapsack

    */
    #include <stdio.h>

        int n;
        int cost[50];
        int value[50];
        int W;
        int q;

    void knapsack_fill() {
        int current_weight;
        float total_value;
        int i, maximum_i;
        int used[10];

        for (i = 0; i < n; ++i)
            used[i] = 0;

        current_weight = W;

        while (current_weight > 0) { // While the bag is NOT full :
add

            // Find the suitable object to ADD

            maximum_i = -1;

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

                if ((used[i] == 0) &&((maximum_i == -1) ||
((float)value[i]/cost[i]                                        >
(float)value[maximum_i]/cost[maximum_i])))
                    maximum_i = i;

            used[maximum_i] = 1; // Maximum value used
            current_weight -= cost[maximum_i];
            total_value += value[maximum_i];

            if (current_weight >= 0)
```

```
                printf("\tObject Added : %d\t VALUE : %d\t PROFIT : %d\t Space Left : %d\n", maximum_i + 1, value[maximum_i], cost[maximum_i], current_weight);

            else {

                printf("\tObject Added : %d\t VALUE : %d\t PROFIT : %d\t Space Left : %d\n", (int)((1 + (float)current_weight/cost[maximum_i]) * 100), value[maximum_i], cost[maximum_i], maximum_i + 1);
                total_value -= value[maximum_i];
                total_value += (1 + (float)current_weight/cost[maximum_i]) * value[maximum_i];
            }

        }
        printf("\n\n");
        printf("Final VALUE in the bag :   %.2f.\n\n", total_value);

    }
    int main(){

        printf("\n\n");

        printf("\tImplementation of Fractional Knapsack Problem\n");

printf("\t================================================\n\n");
        printf("\t   Number of objects : ");
        scanf("%d",&n);

        printf("\t   COST of each object - ");
        for(q=0;q<n;q++){
            scanf("%d",&cost[q]);
        }

        printf("\t   PROFIT on each corresponding object - ");
        for(q=0;q<n;q++){
            scanf("%d",&value[q]);
        }

        printf("\tTotal Weight of the BAG : ");
        scanf("%d",&W);
```

```
    printf("\n\n");
    knapsack_fill();
    return 0;

}
```