

# Knapsack-greedy\_method

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Implement in Java, the 0/1 Knapsack problem using Greedy method

## Input Format

7 15 6 10 18 15 3 5 7 1 2 4 5 1 3 7

## Constraints

--

## Output Format

Net Profit: 55.33333333333336 The objects picked up into knapsack are: 1.0 1.0 1.0 1.0 1.0 0.6666666666666666 0.0

## Sample Input 0

```

7
15
6
10
18
15
3
5
7
1
2
4
5
1
3
7
    
```

## Sample Output 0

```

Net Profit: 55.33333333333336
The objects picked up into knapsack are:
1.0
1.0
1.0
1.0
1.0
0.6666666666666666
0.0
    
```





Contest ends in 9 days

Submissions: 100

Max Score: 10

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆



```
1 import java.util.Scanner;
2 class GKnapsack
3 {
4     int n;
5     double c;
6     double p[];
7     double w[];
8     public GKnapsack(int n,double c,double[] p,double[] w)
9     {
10         super();
11         this.n=n;
12         this.c=c;
13         this.p=p;
14         this.w=w;
15     }
16     void compute()
17     {
18         int i;
19         double[] x=new double[n+1];
20         for(i=0;i<n;i++)
21         {
22             x[i]=0.0;
23         }
24         double rc=c;
25         for(i=0;i<n;i++)
26         {
27             if(w[i]>rc)break;
28             x[i]=1;
29             rc=rc-w[i];
30         }
31         if(i<=n)
32         {
33             x[i]=rc/w[i];
34         }
35         double netProfit=0.0;
36         for(i=0;i<n;i++)
37         {
38             if(x[i]>0.0)
39             {
40                 netProfit=netProfit+x[i]*p[i];
41             }
42         }
43         System.out.println("Net Profit: "+netProfit);
44         System.out.println("The objects picked up into knapsack are:");
45         for(i=0;i<n;i++)
46         {
47             System.out.println(x[i]+" ");
48         }
49     }
50 }
51
52 public class KpGreedy
53 {
54     public static void main(String[] args)
55     {
56         int n;
57         double c;
58         Scanner input=new Scanner(System.in);
59         //System.out.println("Enter number of objects");
60         n=input.nextInt();
61         double[] p=new double[n+1];
62         double[] w=new double[n+1];
63         int i;
64         //System.out.println("Enter capacity of Knapsack");
65         c=input.nextDouble();
66         //System.out.println("Enter profit for each "+n+" objects");
67         for(i=0;i<n;i++)
68         {
69             p[i]=input.nextDouble();
70             //System.out.println("Enter weight for each "+n+" objects");
```

```

70         for(i=0;i<n;i++)
71             w[i]=input.nextDouble();
72         GKnapsack gk=new GKnapsack(n,c,p,w);
73         gk.compute();
74     }
75 }
76
77
78
79
80
81
82

```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)

[Run Code](#)

[Submit Code](#)

Testcase 0 

**Congratulations, you passed the sample test case.**

Click the **Submit Code** button to run your code against all the test cases.

**Input (stdin)**

```

7
15
6
10
18
15
3
5
7
1
2
4
5
1
3
7

```

**Your Output (stdout)**

```

Net Profit: 55.33333333333336
The objects picked up into knapsack are:
1.0
1.0
1.0
1.0
1.0
0.6666666666666666
0.0

```

**Expected Output**

```

Net Profit: 55.33333333333336
The objects picked up into knapsack are:
1.0
1.0
1.0
1.0
1.0
0.6666666666666666
0.0

```