Knapsack-greedy_method

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: 55.333333333333336 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: 55.333333333333336
The objects picked up into knapsack are:
1.0
1.0
1.0
1.0
1.0
0.6666666666666666
0.0
```

Java 7

```
import java.util.Scanner;
class GKnapsack
{
    int n;
    double c;
    double p[];
    double w[];
    public GKnapsack(int n,double c,double[] p,double[] w)
    {
        super();
        this.n=n;
        this.c=c;
        this.p=p;
        this.w=w;
    }
    void compute()
    {
        int i;
        double[] x=new double[n+1];
        for(i=0;i<n;i++)
        {
            x[i]=0.0;

        }
        double rc=c;
        for(i=0;i<n;i++)
        {
            if(w[i]>rc)break;
            x[i]=1;
            rc=rc-w[i];
        }
        if(i<=n)
        {
            x[i]=rc/w[i];
        }
        double netProfit=0.0;
        for(i=0;i<n;i++)
        {
            if(x[i]>0.0)
            {
                netProfit=netProfit+x[i]*p[i];
            }
        }
        System.out.println("Net Profit: "+netProfit);
        System.out.println("The objects picked up into knapsack are:");
        for(i=0;i<n;i++)
        {
            System.out.println(x[i]+" ");
        }
    }
}
public class KpGreedy
{
    public static void main(String[] args)
    {
        int n;
        double c;
        Scanner input=new Scanner(System.in);
        //System.out.println("Enter number of objects");
        n=input.nextInt();
        double[] p=new double[n+1];
        double[] w=new double[n+1];
        int i;
        //System.out.println("Enter capacity of Knapsack");
        c=input.nextDouble();
        //System.out.println("Enter profit for each "+n+" objects");
        for(i=0;i<n;i++)
            p[i]=input.nextDouble();
        //System.out.println("Enter weight for each "+n+" objects");
```

```
        for(i=0;i<n;i++)
            w[i]=input.nextDouble();
        GKnapsack gk=new GKnapsack(n,c,p,w);
        gk.compute();
    }
}
```

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: 55.333333333333336
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: 55.333333333333336
The objects picked up into knapsack are:
1.0
1.0
1.0
1.0
1.0
0.6666666666666666
0.0
```