NAME: SHUBHAM TAKANKHAR

CLASS: SY MCA

ROLL NO: 54

ASSIGNMENT 3: Write program to solve fractional knapsack using Greedy algoritmn.

SOURCE CODE:

<<KnapSack Problem:Maximization>>

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

public class knapSack {
    ArrayList<ArrayList<Integer>> items = new ArrayList<ArrayList<Integer>>();
    int knapSackSize;
    void getData(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Size of knapSack:");
        knapSackSize = sc.nextInt();
        System.out.println("Enter no. of items:");
        int itemNos = sc.nextInt();

        for (int i = 0; i < itemNos; i++){
            System.out.println("Enter Item No."+(i+1));
            System.out.println("Enter Value");
            int val = sc.nextInt();
            System.out.println("Enter Weight");
            int weight = sc.nextInt();
            ArrayList<Integer> pair = new ArrayList<Integer>();
            pair.add(val);
            pair.add(weight);
            int ratio = val / weight;
            pair.add(ratio);
            items.add(pair);

        }
        sc.close();
    }
```

```
    void printRawData(){
        System.out.printf("Raw Data of %d Items [value,weight,ratio]: ",items.siz
e());
        for (ArrayList<Integer> item:items){
            System.out.print(item+" ");
        }
    }

    void sortItems(){
        Collections.sort(items, new Comparator<ArrayList<Integer>>(){
            @Override
            public int compare(ArrayList<Integer> o1, ArrayList<Integer> o2){
                return o1.get(2).compareTo(o2.get(2));
            }
        }.reversed());
    }

    void maxiMize(){
        int maxValue = 0;
        int totalWeight = knapSackSize;
        for (ArrayList<Integer> item : items){
            if (totalWeight > item.get(1)){
                totalWeight -= item.get(1); //40
                maxValue += item.get(0);
            }
            else if (totalWeight>0){
                maxValue += totalWeight * item.get(2);
                totalWeight = 0;
            }
            else{
                break;
            }
        }
        System.out.println("\nMaximum Value Obtained Is: " + maxValue);
    }
    public static void main(String[] args){
        knapSack sack = new knapSack();
        sack.getData();
        sack.sortItems();
        sack.printRawData();
        sack.maxiMize();
    }
}
```

OUTPUT:

```
Shubham@N073B00K ~
❯ & 'c:\Users\Shubham\.vscode\extensions\vscjava.vscode-java-debug-0.32.1\scripts\launcher.bat'
ailsInExceptionMessages' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\Shubham\AppData\Local\Temp\vscod
Enter Size of knapSack:
50
Enter no. of items:
4
Enter Item No.1
Enter Value
80
Enter Weight
10
Enter Item No.2
Enter Value
100
Enter Weight
20
Enter Item No.3
Enter Value
130
Enter Weight
30
Enter Item No.4
Enter Value
120
Enter Weight
40
Raw Data of 4 Items [value,weight,ratio]: [80, 10, 8] [100, 20, 5] [130, 30, 4] [120, 40, 3]
Maximum Value Obtained Is: 260
Shubham@N073B00K ~
❯
```