import java.lang.\*;

import java.util.Arrays;

import java.util.Comparator;

import java.util.Scanner;

// Greedy approach

public class FractionalKnapSack {

// Function to get maximum value

private static double getMaxValue(ItemValue[] arr, int capacity) {

// Sorting items by profit/weight ratio;

Arrays.sort(arr, new Comparator<ItemValue>() {

@Override

public int compare(ItemValue item1, ItemValue item2) {

double cpr1 = (double) item1.profit / item1.weight;

double cpr2 = (double) item2.profit / item2.weight;

if (cpr1 < cpr2)

return 1;

else

return -1;

}

});

double totalValue = 0d;

for (ItemValue i : arr) {

int curWt = i.weight;

int curVal = i.profit;

if (capacity - curWt >= 0) {

// This weight can be picked whole

capacity = capacity - curWt;

totalValue += curVal;

} else {

// Item cant be picked whole

double fraction = (double) capacity / curWt;

totalValue += (curVal \* fraction);

capacity = (int) (capacity - (curWt \* fraction));

break;

}

}

return totalValue;

}

// Item value class

static class ItemValue {

int profit, weight;

// Item value function

public ItemValue(int val, int wt) {

this.weight = wt;

this.profit = val;

}

}

// Driver code

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of items: ");

int n = scanner.nextInt();

ItemValue[] arr = new ItemValue[n];

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

System.out.println("Enter profit and weight for item " + (i + 1) + ": ");

int profit = scanner.nextInt();

int weight = scanner.nextInt();

arr[i] = new ItemValue(profit, weight);

}

System.out.println("Enter the capacity of the knapsack: ");

int capacity = scanner.nextInt();

double maxValue = getMaxValue(arr, capacity);

// Function call

System.out.println("Maximum value we can obtain = " + maxValue);

}

}