**快乐司机**

问题描述

　　"嘟嘟嘟嘟嘟嘟  
　　喇叭响  
　　我是汽车小司机  
　　我是小司机  
　　我为祖国运输忙  
　　运输忙"  
　　这是儿歌“快乐的小司机”。话说现在当司机光有红心不行，还要多拉快跑。多拉不是超载，是要让所载货物价值最大，特别是在当前油价日新月异的时候。司机所拉货物为散货，如大米、面粉、沙石、泥土......  
　　现在知道了汽车核载重量为w，可供选择的物品的数量n。每个物品的重量为gi,价值为pi。求汽车可装载的最大价值。（n<10000,w<10000,0<gi<=100,0<=pi<=100)

输入格式

　　输入第一行为由空格分开的两个整数n w  
　　第二行到第n+1行，每行有两个整数，由空格分开，分别表示gi和pi

输出格式

　　最大价值（保留一位小数）

样例输入

5 36  
99 87  
68 36  
79 43  
75 94  
7 35

样例输出

71.3  
解释：  
先装第5号物品，得价值35，占用重量7  
再装第4号物品，得价值36.346,占用重量29  
最后保留一位小数，得71.3

C++代码：

#include<stdio.h>

#include<algorithm>

using namespace std;

struct stt{

int g,p;

double v;

bool operator<(const stt & t)const{

return v>t.v;

}

}a[10005];

int main()

{

int n,w;

scanf("%d%d",&n,&w);

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

{

scanf("%d%d",&a[i].g,&a[i].p);

a[i].v=1.0\*a[i].p/a[i].g;

}

sort(a,a+n);

double sum=0.0;

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

{

if(w>0){

if(w>a[i].g) w-=a[i].g,sum+=a[i].p;

else sum+=w\*a[i].v,w=0;

}

else break;

}

printf("%.1lf\n",sum);

return 0;

}

C代码：

#include <stdio.h>

#include <stdlib.h>

struct st{

int g;

int p;

double zhi;

}stt[10005];

int cmp(const void \*a,const void \*b);

int main(){

int n,w,i;

double sum=0.0;

scanf("%d%d",&n,&w);

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

scanf("%d %d",&stt[i].g,&stt[i].p);

stt[i].zhi=stt[i].p\*1.0/stt[i].g;

}

qsort(stt,n,sizeof(stt[0]),cmp);

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

if(w>=stt[i].g){

sum+=stt[i].p;

w-=stt[i].g;

}else{

sum+=stt[i].zhi\*w;

break;

}

if(w<=0){

break;

}

}

printf("%.1lf\n",sum);

return 0;

}

int cmp(const void \*a,const void \*b){

struct st \*aa=(struct st \*)a;

struct st \*bb=(struct st \*)b;

return (aa->zhi)<(bb->zhi)?1:-1;

}

JAVA代码：

import java.util.\*;

import java.io.\*;

class Node implements Comparable<Node>{

double g;

double p;

public Node(double g,double p){

this.g = g;

this.p = p;

}

public int compareTo(Node n){

double a = this.p/this.g;

double b = n.p/n.g;

if(a<b) return 1;

else return -1;

}

}

public class Main{

public static void main(String[] args)throws IOException{

StreamTokenizer st = new StreamTokenizer(new BufferedReader(new InputStreamReader(System.in)));

PriorityQueue<Node> q = new PriorityQueue<>();

st.nextToken(); int n = (int)st.nval;

st.nextToken(); double w = st.nval;

double a=0,b=0;

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

st.nextToken();

a = st.nval;

st.nextToken();

b = st.nval;

q.add(new Node(a,b));

}

// while(!q.isEmpty()){

// System.out.println(q.poll().g);

// }

int k = 1;

double sum = 0;

while(w>0){

Node tmp = q.poll();

if(w>=tmp.g){

sum += tmp.p;

}else{

sum += tmp.p/tmp.g\*w;

}

w = w - tmp.g;

k++;

if(k==n+1) break;

}

System.out.printf("%.1f\n", sum);

}

}