

# C4-A题题解

作者：余绍函

## 题目描述

Abyss 有一次和同学聚餐，餐厅上了许许多多美味的佳肴，Abyss 「。●\_●。」地很开心。

Abyss 嘴里默念着：「我现在一共有  $k$  的胃口，桌子上有  $n$  道菜，第  $i$  道菜的量为  $w_i$ ，其美味值为  $v_i$ ，如果吃掉  $j$  个单位( $0 < j \leq w_i$ ) 且  $j$  为整数) 的第  $i$  道菜，那么我能获得的美味值是  $j * v_i / w_i$ ，我最终的收益是全部吃到的菜的美味值之和，我能够吃掉的菜的总量不能超过我的最大胃口  $k$ ，那我最终可以获得的最大美味值是多少呢？」

Abyss 其实在瞬间就计算完毕了，但是他不确定是不是正确的(´\_` )，于是他希望你也帮忙计算一下他能获得的最大美味值是多少。

## 输入格式

第一行两个正整数  $n, k$  ( $1 \leq n \leq 10^3, 1 \leq k \leq 10^5$ )，表示上菜的数量和总胃口  $k$ 。

接下来  $n$  行，每行两个正整数  $v_i, w_i$  ( $1 \leq v_i \leq 10^7, 1 \leq w_i \leq 10^6$ )，表示第  $i$  道菜的美味值和量。

## 输出格式

一行一个保留三位小数的实数，表示可以获得的最大美味值。

## 输入样例

```
3 6
1 1
1 3
2 4
```

## 输出样例

```
3.333
```

# 样例解释

一种最优方案为吃掉全部的第 1 道菜、 $1/2$  第 2 道菜和 全部的第 3 道菜。

## Hint

**大胆贪心，小心求证！**

经典贪心问题，本题具体可以参考算法导论 P244 或者课件 PPT。

## 解题思路

既然我可以不把一道菜都吃完，那么我肯定先吃 美味率（美味率=美味值/量） 最高的的菜啊！那么这道题就变成了一个简单的排序了，话不多说，上代码：

```
#include<bits/stdc++.h>
using namespace std;
int n, ind = 0;
typedef struct c {
    double v, k;
}dish;
double k, v[1005], w[1005], value = 0;
dish dishes[1005];
int cmp(dish *p1, dish *p2) {
    return ((dish *)p1) -> v < ((dish *)p2) -> v;
}
int main() {
    scanf("%d%lf", &n, &k);
    for(int i = 0; i < n; i++) {
        scanf("%lf%lf", &(dishes[i].v), &(dishes[i].k));
        dishes[i].v /= dishes[i].k;
    } sort(dishes, dishes + n, [](const dish& p1, const dish& p2) {return p1.v > p2.v;});
    while(k > 0 && ind < n) {
        if(dishes[ind].k <= k) {
            value += (dishes[ind].v * dishes[ind].k);
            k -= dishes[ind].k;
        } else {
            value += (dishes[ind].v * k);
            k = 0;
        } ind++;
    } printf("%.3lf", value);
```

```
return 0;
```

```
}
```