


## 实验六

 编程题 1 (总分: 100.00)

#	题目	分值
1	实验6-1: Knapsack Problem	100.00

## Description

给定 $n$ 种物品和一个背包。物品 $i$ 的重量是 $W_i$ ，其价值为 $V_i$ ，背包的容量为 $C$ 。应如何选择装入背包的物品，使得装入背包中物品的总价值最大？在选择物品 $i$ 装入背包时，可以选择物品 $i$ 的一部分，而不一定要全部装入背包， $1 \leq i \leq n$ 。

## 编程任务：

对于给定的 $n$ 种物品和一个背包容量 $C$ ，编程计算装入背包中最大的物品总价值。

## Input

输入由多组测试数据组成。

每组测试数据输入的第1行中有2个正整数 $n$ 和 $C$ 。正整数 $n$ 是物品个数；正整数 $C$ 是背包的容量。接下来的2行中，第一行有 $n$ 个正整数，分别表示 $n$ 个物品的重量，它们之间用空格分隔；第二行有 $n$ 个正整数，分别表示 $n$ 个物品的价值，它们之间用空格分隔。

## Output

对应每组输入，输出的每行是计算出的装入背包中最大的物品总价值，保留一位有效数字。

## Sample Input

```
3 50
10 20 30
60 100 120
```

## Sample Output

```
240.0
```

参考答案:

```
#include<iostream>
#include<stdio.h>
#include<algorithm>
using namespace std;
struct Node
{
    double weight;
    double value;
};
bool comp (Node a,Node b)
{
    return a.value/a.weight > b.value/b.weight;
}
int main()
{
    int n,i;
    double c,w,v;
    while(scanf("%d%lf",&n,&c)!=EOF)
    {
        Node infor[2001];
        for(i=0;i<n;i++)
            scanf("%lf",&infor[i].weight);
        for(i=0;i<n;i++)
            scanf("%lf",&infor[i].value);
        sort(infor,infor+n,comp);
        v = 0.0;
        w = 0.0;
        for(i=0;i<n;i++)
        {
            if(infor[i].weight + w <= c)
            {
                w += infor[i].weight;
                v += infor[i].value;
            }
            else
            {
                v+=(c-w)/infor[i].weight
* infor[i].value;
                break;
            }
        }
        printf("%.1lf\n",v);
    }
    return 0;
}
```

