

## 1242：网线主管

## 题目描述

仙境的居民们决定举办一场程序设计区域赛。裁判委员会完全由自愿组成，他们承诺要组织一次史上最公正的比赛。他们决定将选手的电脑用星形拓扑结构连接在一起，即将它们全部连到一个单一的中心服务器。为了组织这个完全公正的比赛，裁判委员会主席提出要将所有选手的电脑等距离地围绕在服务器周围放置。

为购买网线，裁判委员会联系了当地的一个网络解决方案提供商，要求能够提供一定数量的等长网线。裁判委员会希望网线越长越好，这样选手们之间的距离可以尽可能远一些。

该公司的网线主管承接了这个任务。他知道库存中每条网线的长度（精确到厘米），并且只要告诉他所需的网线长度（精确到厘米），他都能够完成对网线的切割工作。但是，这次，所需的网线长度并不知道，这让网线主管不知所措。

你需要编写一个程序，帮助网线主管确定一个最长的网线长度，并且按此长度对库存中的网线进行切割，能够得到指定数量的网线。

## 输入

第一行包含两个整数 $N$ 和 $K$ ，以单个空格隔开。 $N$  ( $1 \leq N \leq 10000$ ) 是库存中的网线数， $K$  ( $1 \leq K \leq 10000$ ) 是需要的网线数量。

接下来 $N$ 行，每行一个数，为库存中每条网线的长度（单位：米）。所有网线的长度至少1m，至多100km。输入中的所有长度都精确到厘米，即保留到小数点后两位。

## 输出

网线主管能够从库存的网线中切出指定数量的网线的最长长度（单位：米）。必须精确到厘米，即保留到小数点后两位。

若无法得到长度至少为1cm的指定数量的网线，则必须输出“0.00”（不包含引号）。

## 输入样例

```
4 11
8.02
7.43
4.57
5.39
```

## 输出样例

```
2.00
```

## 分析

本题同样也是一道二分搜索，我们需要在不同的网线长度中找到能够裁剪出指定数量的数值。

我们首先需要考虑这道题的两个边界，最大的边界很容易考虑到，那就是最长的那根绳子。那么最小的边界应该设置为几呢？0还是1（厘米），答案应该是1。

如果设置为0的话，当给定的电线不够切出所需的长度时，则会出现异常。（等待补充测试数据）

因此，必须将最小值设置为1。并在最后输出结果时，判断一下最后的结果是否满足需求。

此外，本题依然是一道浮点数的运算，注意转化成整数。由于本题的结果不是四舍五入，所以在计算精度的时候，只需要乘以100即可。同理，mid的变化也是加1和减1。

第三，本题在第一次找到符合的数量后，千万不要急于终止，要记得我们是尽量选择更加长的绳子，参考前面的课程中求边界索引的做法，此刻我们需要将左边界右移，继续尝试更大的数值。

## 编码

```
#include<bits/stdc++.h>

using namespace std;
//网线最大数量是10000
int const N = 1e4 + 1;
//防止数据移除
long long n, s;
//存储库存网线的长度
double b[N];

//统计在x的长度下，最多能够截取多少网线
long long ts(double x) {
    long long int num = 0;
    for (int i = 1; i <= n; i++) {
        num += b[i] / x;
    }
    return num;
}

//二分搜索
double Search(int l, int r) {
    int mid;
    while (l <= r) {
        mid = l + (r - l) / 2;
        long long num = ts(mid);
        //尽量选择更长的
        if (num == s) {
            l = mid + 1;
        }
        //绳子切的短了，左边界增加
        else if (num > s) {
            l = mid + 1;
        }
        //绳子切的长了，右边界减少
        else if (num < s) {
            r = mid - 1;
        }
    }
    return l;
}
```

```

        r = mid - 1;
    }
}
//尽可能选择更长的绳子
return r;
}

int main() {
    cin >> n >> s;
    int maxr = 0;
    for (int i = 1; i <= n; i++) {
        double len;
        cin >> len;
        //将小数转化为整数
        b[i] = len * 100;
        if (b[i] > maxr)
            maxr = b[i];
    }
    double res = Search(1, maxr);
    //判断一下最终的结果是否小于1厘米
    if (res < 1) {
        printf("%.2lf", 0);
    } else {
        printf("%.2lf", res / 100);
    }
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

