

## P2440 木材加工

### 题目描述

木材厂有  $n$  根原木，现在想把这些木头切割成  $k$  段长度均为  $l$  的小段木头（木头有可能有剩余）。

当然，我们希望得到的小段木头越长越好，请求出  $l$  的最大值。

木头长度的单位是  $\text{cm}$ ，原木的长度都是正整数，我们要求切割得到的小段木头的长度也是正整数。

例如有两根原木长度分别为 11 和 21，要求切割成等长的 6 段，很明显能切割出来的小段木头长度最长为 5。

### 输入格式

第一行是两个正整数  $n, k$ ，分别表示原木的数量，需要得到的小段的数量。

接下来  $n$  行，每行一个正整数  $L_i$ ，表示一根原木的长度。

### 输出格式

仅一行，即  $l$  的最大值。

如果连  $1\text{cm}$ 长的小段都切不出来，输出 0。

### 输入样例

```
3 7
232
124
456
```

### 输出样例

```
114
```

### 解析

经典的二分搜索，将长度在1和最大长度之间进行搜索，直到裁切的段数等于目标数。

### 编码

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

```

using namespace std;
//网线最大数量是100000
long long const N = 1e5 + 1;
//防止数据移除
long long n, s;
//存储库存网线的长度
long long b[N];

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

//二分搜索
long long Search(long long l, long long r) {
    long long 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;
        }
    }
    //尽可能选择更长的绳子
    return r;
}

int main() {
    cin >> n >> s;
    long long maxr = 0;
    for (int i = 1; i <= n; i++) {
        cin >> b[i];
        if (b[i] > maxr) {
            maxr = b[i];
        }
    }
}

```

```
}  
long long res = Search(1, maxr);  
//判断一下最终的结果是否小于1厘米  
if (res < 1) {  
    printf("%ld", 0);  
} else {  
    printf("%ld", res);  
}  
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。

