一些注意点和试题选讲

—— 郑任毅

C 还是 C++？

- 一个简单的问题：在做 PAT 题目的时候，选择 C 还是 C++？ 如果没有学过 C++，选哪个?

C 还是 C++？

- 一个简单的问题：在做 PAT 题目的时候，选择 C 还是 C++？ 如果没有学过 C++ ，选哪个?
- 毫无疑问，我们始终推荐 C++ ，即使我的代码中完全没有一点 C++ 的语法特性，也不使用 cin/cout 。
- C++ 完全兼容所有 C 的语法，因此对于没有学过 C++ 的同学来说，也只需要将文件后缀改成 .cpp 、提交的时候选择 g++ 就可以了。

C 还是 C++？

- 一个简单的问题：在做 PAT 题目的时候，选择 C 还是 C++？ 如果没有学过 C++ ，选哪个?
- C++ 的优点：
  - 变量定义的灵活性，至少并不强制定义在语句块的开头
  - 编译错误会变少（大概会）
  - 更方便使用 algorithm 头文件以及其他 STL 库
  - 更多可选用的语法

关于数据范围和数组大小

- 数组究竟要定义多大?

关于数据范围和数组大小

- 数组究竟要定义多大?
- 事实上这最终取决于你可能会用到多少。

```
int n;
scanf("%d",&n);
for(int i = 1 ; i <= n ; ++i){
    scanf("%d", &a[i]);
}
```

这样一个代码， a数组应该开多大?

关于数据范围和数组大小

- 数组究竟要定义多大?
- 事实上这最终取决于你可能会用到多少。

```
int n;
scanf("%d",&n);
for(int i = 1 ; i <= n ; ++i){
    scanf("%d", &a[i]);
}
```

这样一个代码， a 数组应该开多大?

题面描述中 n 的范围是多大， a 就该开至少比这个范围多 1 的大小

关于数据范围和数组大小

- 数组究竟要定义多大?
- 10 的 5 次方?
  - 10005？
  - 少了一个 0！！！

关于数据范围和数组大小

- 数组究竟要定义多大?
- 10 的 5 次方?
  - 10005？
  - 少了一个 0！！！

```
const  int  maxn = 1e5 + 5;
int a[maxn];
```

关于数据范围和数组大小

- 数组究竟要定义多大?
- 不超过 80 个字符的字符串?
  - char str[80]?
  - 读入字符串时最后会有一个额外的 '\0' ！！！
  - char str[85];

%c 读字符 & scanf 格式化读空格?

- 需要读入单个字符?
  - 用 scanf(“%c”,&ch); ?
- 读入两个数，中间以空格间隔?
  - 用 scanf(“%d %d”,&a,&b); ?
- 读入两个字符，中间以空格间隔?
  - 用 scanf(“%c %c”,&ch1,&ch2); ?

%c 读字符 & scanf 格式化读空格?

- 需要读入单个字符?
  - 用 scanf(“%c”,&ch); ?
- 读入两个数，中间以空格间隔?
  - 用 scanf(“%d %d”,&a,&b); ?
- 读入两个字符，中间以空格间隔?
  - 用 scanf(“%c %c”,&ch1,&ch2); ?
- 以上均坚决不推荐！！！

%c 读字符 & scanf 格式化读空格?

- 需要读入单个字符?

```
char str[5];
scanf("%s",str);
char ch = str[0];
```

%c 读字符 & scanf 格式化读空格?

- 读入两个数，中间以空格间隔?

  scanf("%d%d",&a,&b);

- 同理，读一个数，一个字符串，以空格间隔?

  scanf("%d%s",&a,str);

- 读一个字符串，一个数，以空格间隔?

  scanf("%s%d",str,&a);

- 读两个字符串，以空格间隔?

  scanf("%s%s",str1,str2);

%c 读字符 & scanf 格式化读空格?

- 读入两个字符，中间以空格间隔?

```
char str1[5], str2[5];
scanf("%s%s",str1,str2);
char ch1 = str1[0], ch2 = str2[0];
```

- 读入一个数字，一个字符，中间以空格间隔?

```
char str[5];
scanf("%d%s",&a,str);
char ch = str[0];
```

%c 读字符 & scanf 格式化读空格?

- 总而言之， scanf 的读入机制是什么?
- 1 、除了 %c 以外，对于所有 %d 、 %s 、 %lf 、 %lld 等，如果他们在 format 串中紧挨着没有任何分隔符，那么在读入时会忽略读入两个值中间的所有空格、回车、制表符
- 2 、当任意占位符之间有其他分隔符，包括空格、冒号等，那么将在读完第一个值之后严格匹配掉占位符，再读入第二个值
- 3 、 %c 会读入任意一个字符，包括空格等

%c 读字符 & scanf 格式化读空格?

- scanf 的上述用法不能解决什么问题?
- 行级的字符串，并且中间可能有空格或者制表符
- 这时需要用 gets(s)
- 注意 gets(s) 的上一行，如果是用 scanf 读入，那么可以在 scanf 后、 gets(s) 前再额外加一个 gets(s) ，用于读掉上一行的行末回车。

字符串中的数读入

- 在一个已经存入字符串数组的字符串中，有一部分是一个整数，我该如何方便地将它读出来?
  - 如 100/15　3E-10

字符串中的数读入

- 在一个已经存入字符串数组的字符串中，有一部分是一个整数，我该如何方便地将它读出来?
  - 如 100/15　3E-10
- 在此，我提供一种通用的读入方式，即找到这个数在字符串 s 中的开始和结束位置（即 s[l, r] 是这个数），那么我就可以方便地读入。

字符串中的数读入

```
int getNum(char s[], int l, int r){
    int ans = 0, d = 1;
    if(s[l] == '-')d = -1, l++;
    if(s[l] == '+')d = 1, l++;
    for(int i = l ; i <= r ; ++i){
        ans = ans * 10 + s[i] - '0';
    }
    return d * ans;
}
```

变量 ans 是绝对值， d 是符号

首先判断第一位是不是符号

然后对剩余位依次遍历，获得结果

例题 B1024 科学计数法

- 科学计数法是科学家用来表示很大或很小的数字的一种方便的方法，其满足正则表达式 [+-][1-9]"."[0-9]+E[+-][0-9]+ ，即数字的整数部分只有 1 位，小数部分至少有 1 位，该数字及其指数部分的正负号即使对正数也必定明确给出。
- 现以科学计数法的格式给出实数 A，请编写程序按普通数字表示法输出 A，并保证所有有效位都被保留。
- 输入格式：每个输入包含 1 个测试用例，即一个以科学计数法表示的实数 A。该数字的存储长度不超过 9999 字节，且其指数的绝对值不超过 9999。
- 输出格式：对每个测试用例，在一行中按普通数字表示法输出 A，并保证所有有效位都被保留，包括末尾的 0。

例题 B1024 科学计数法

- 输入样例 1：
- +1.23400E-03
- 输出样例 1：
- 0.00123400
- 输入样例 2：
- -1.2E+10
- 输出样例 2：
- -12000000000

例题 B1024 科学计数法

- 思路：对于一个 a E b 形式的实数，该如何显示

例题 B1024 科学计数法

- 思路：对于一个 a E b 形式的实数，该如何显示
- 重点：要加几个零？小数点加哪？

例题 B1024 科学计数法

- 思路：对于一个 a E b 形式的实数，该如何显示
- 重点：要加几个零？小数点加哪？

+1.23400E-03
0.00123400

b 为负数时

在 a 串前加 |b| 个 0

小数点在第一个 0 后

-1.2E+10
-12000000000

b 为正数时

a 串尾加 0 至 b+1 位

要小数点吗？

例题 B1024 科学计数法

- 思路：对于一个 a E b 形式的实数，该如何显示
- 重点：要加几个零？小数点加哪？

+1.23400E-03
0.00123400

b 为负数时

在 a 串前加 |b| 个 0

小数点在第一个 0 后

-1.2E+10
-12000000000

b 为正数时

a 串尾加 0 至 b+1 位

要小数点吗？

若有效数字大于 b+1

则在 b+1 位后加小数点

例题 B1024 科学计数法

- 思路：对于一个 a E b 形式的实数，该如何显示
- 重点：要加几个零？小数点加哪？
- b 为 0 呢？
  - 虽然不知道有没有这样的情况
  - 但是我们发现可以归于 b 为正数一类
- 记得输出负号（如果是负数）

例题 B1024 科学计数法

- 于是我们发现：
  - 对于 a 我们并不关心它中间的小数点，只需要存下它出现的数字
  - 对于 b，我们可以用字符串中读数的方法，读出它的值
  - 如果是负数，输出负号
  - 根据之前对 b 的正负情况分类讨论，可以分别处理输出

例题 B1024 科学计数法

- 读入与预处理

```
scanf("%s",s);
string a;
int d,b,pos;
if(s[0] == '-')d = -1;
else d = 1;
for(int i = 1 ; s[i] ; ++i){
    if(s[i] == 'E'){
        pos = i;
        break;
    }
    if(s[i] >= '0' && s[i] <= '9'){
        a += s[i];
    }
}
b = getNum(s, pos + 1, strlen(s) - 1);
```

例题 B1024 科学计数法

- 分类讨论并输出

```
if(d == -1)printf("-");

if(b < 0){
    for(int i = 1 ; i <= -b ; ++i)a = '0' + a;
    for(int i = 0 ; i < a.length() ; ++i){
        printf("%c",a[i]);
        if(i == 0)printf(".");
    }
}
else{
    if(a.length() < b + 1){
        int add = b + 1 - a.length();
        for(int i = 1 ; i <= add ; ++i)a += '0';
    }
    for(int i = 0 ; i < a.length() ; ++i){
        printf("%c",a[i]);
        if(a.length() > b + 1 && i == b)printf(".");
    }
}
printf("\n");
```

例题 B1024 科学计数法

- 我个人的另一种写法

```
if(b < 0){
    for(int i = 1 ; i <= -b ; ++i)a = '0' + a;
}

if(d == -1)printf("-");
for(pos = 0 ; pos < a.length() ; ++pos){
    printf("%c",a[pos]);
    if(pos == 0 && a[pos] == '0')printf(".");
    if(pos == b && pos != a.length() - 1)printf(".");
}
for(pos; pos <= b; ++pos)printf("0");
printf("\n");
```

数值范围

- char : [-127 , 128]
- 不要用 char 作为计数器
- int : - 2e9 ~ 2e9
- 1e5 × 1e5 就会超过 int

例题 B1034 有理数四则运算

- 本题要求编写程序，计算 2 个有理数的和、差、积、商。
- 输入格式：
- 输入在一行中按照 "a1/b1 a2/b2" 的格式给出两个分数形式的有理数，其中分子和分母全是整型范围内的整数，负号只可能出现在分子前，分母不为 0。
- 输出格式：
- 分别在 4 行中按照 " 有理数 1 运算符 有理数 2 = 结果 " 的格式顺序输出 2 个有理数的和、差、积、商。注意输出的每个有理数必须是该有理数的最简形式 "k a/b"，其中 k 是整数部分， a/b 是最简分数部分；若为负数，则须加括号；若除法分母为 0，则输出 "Inf"。题目保证正确的输出中没有超过整型范围的整数。

例题 B1034 有理数四则运算

输入样例 1：

2/3 -4/2

输出样例 1：

2/3 + (-2) = (-1 1/3)
2/3 - (-2) = 2 2/3
2/3 * (-2) = (-1 1/3)
2/3 / (-2) = (-1/3)

输入样例 2：

5/3 0/6

输出样例 2：

1 2/3 + 0 = 1 2/3
1 2/3 - 0 = 1 2/3
1 2/3 * 0 = 0
1 2/3 / 0 = Inf

例题 B1034 有理数四则运算

- 思路：这道题的题目描述非常简单，而解题也无非就是读入、计算、输出三个部分。

例题 B1034 有理数四则运算

- 我们首先考虑输出
- 虽然四则运算结果可能都不一样，但是对于有理数的输出格式都是一样的。

例题 B1034 有理数四则运算

- 我们首先考虑输出
- 虽然四则运算结果可能都不一样，但是对于有理数的输出格式都是一样的。
- 我们可以考虑将输出全部统一起来，也就是写一个函数来输出，我们应该传入这个有理数，在函数内格式化并输出出来。

例题 B1034 有理数四则运算

- 研究一下一个有理数应该如何输出：
- 首先需要考虑哪些?
  - 是否为 0
  - 是否为整数
  - 是否为负数
  - 是否要化简
  - 是否是假分数要换成带分数

例题 B1034 有理数四则运算

- 研究一下一个有理数应该如何输出：
- 那么如何依次处理?
  - 分子是 0 直接输出 0
  - 是负数在前后加上 () 和 -
  - 分子是分母的整数倍则为整数，输出数值
  - 分子分母 gcd 不为 1，则需要分子分母同除 gcd 化简
  - 分子大于分母则需要带分数，否则是真分数

例题 B1034 有理数四则运算

- 那么这个函数我们需要什么参数?
  - 正负情况
  - 分子
  - 分母
  - 分子分母都保证是正，并且可以是未化简的情况

例题 B1034 有理数四则运算

```
void print(int d, int a, int b){
    if(a == 0)printf("0");
    else{
        int g = __gcd(a, b);
        a /= g;
        b /= g;
        if(d == -1)printf("(-");
        if(a % b == 0)printf("%d",a/b);
        else{
            if(a > b)printf("%d %d/%d",a/b,a%b,b);
            else printf("%d/%d",a,b);
        }
        if(d == -1)printf(")");
    }
}
```

例题 B1034 有理数四则运算

- 此时我们对已知分子分母和符号的数可以直接调用函数输出
- 然后我们考虑读入
- “a1/b1 a2/b2”
- 负号只可能出现在分子前

例题 B1034 有理数四则运算

- 此时我们对已知分子分母和符号的数可以直接调用函数输出
- 然后我们考虑读入
- “a1/b1 a2/b2”
- 负号只可能出现在分子前
- 我们可以用字符串中读数的方法，但我们也可以直接将负号归在 a 里，认为 a 可正可负，这样就适合格式化读入了
- scanf("%d/%d%d/%d",&a1,&b1,&a2,&b2);

例题 B1034 有理数四则运算

- scanf("%d/%d%d/%d",&a1,&b1,&a2,&b2);
- 在读入完后，将 a1 、 a2 的符号再提取出来，单独存下就可以调用输出函数格式化输出他们了。

例题 B1034 有理数四则运算

- scanf("%d/%d%d/%d",&a1,&b1,&a2,&b2);
- 在读入完后，将 a1 、 a2 的符号再提取出来，单独存下就可以调用输出函数格式化输出他们了。

```
int d1 = 1, d2 = 1;
int a1, b1, a2, b2;
scanf("%d/%d%d/%d",&a1,&b1,&a2,&b2);
if(a1 < 0)d1 = -1, a1 = -a1;
if(a2 < 0)d2 = -1, a2 = -a2;
```

例题 B1034 有理数四则运算

- 接下来我们处理运算：
- 加减方面，由于我们的输出函数可以接受没有化简的分子分母，所以我们可以大胆的通分并计算
- a1/b1 + a2/b2

  = (a1*b2)/(b1*b2) + (a2*b1)/(b1*b2)

  = (a1*b2 + a2*b1)/(b1*b2)
- a1/b1 - a2/b2

  = (a1*b2)/(b1*b2) - (a2*b1)/(b1*b2)

  = (a1*b2 - a2*b1)/(b1*b2)

例题 B1034 有理数四则运算

- 接下来我们处理运算：
- 当然我们还需要加上符号
- d1*a1/b1 + d2*a2/b2

  = (d1*a1*b2 + d2*a2*b1)/(b1*b2)

  分子： d1*a1*b2 + d2*a2*b1　分母： b1*b2
- d1*a1/b1 - d2*a2/b2

  = (d1*a1*b2 – d2*a2*b1)/(b1*b2)

  分子： d1*a1*b2 – d2*a2*b1　分母： b1*b2
- 我们再将分子的符号提取出来，参数传入输出函数就能输出了

例题 B1034 有理数四则运算

- 接下来我们处理运算：
- 乘除方面，我们可以直接计算
- (a1/b1) × (a2/b2)

  = (a1*a2)/(b1*b2)
- (a1/b1) / (a2/b2)

  = (a1*b2)/(a2*b1)
- 需要传入的符号就是 d1*d2
- 需要特别注意的是如果除数为 0 （即 a2 为 0），那么不调用输出函数，直接输出 Inf

例题 B1034 有理数四则运算

- 加法运算的示例：

```
print(d1, a1, b1);
printf(" + ");
print(d2, a2, b2);
printf(" = ");
ans = d1*a1*b2 + d2*a2*b1;
if(ans < 0){
      signal = -1;
      ans = -ans;
}
else signal = 1;
print(signal,ans,b1*b2);
printf("\n");
```

例题 B1034 有理数四则运算

- 这样似乎很合理？但是结果并不如意

时间	结果	得分	题目
9月04日 21:16	部分正确	14	1034

测试点

测试点	结果	用时(ms)
0	答案正确	6
1	答案正确	3
2	答案错误	3
3	浮点错误	8

查看代码

例题 B1034 有理数四则运算

- 浮点错误？一般在除数为 0，特殊运算为负数这类数值问题上才会出现。
- 我认为可能是两个 int 型的数相乘超过了 int 范围导致了这一问题，因为做的时候我简单地认为 PAT 的 B 组题应该不会出现超过 int 范围的情况。
- 于是我用了一个小 Trick 。

欺骗提交

- 竞赛中偶尔会使用的一种手段，将已知是错误的代码提交，用于测试某些数据是否满足预期，一般在数据满足某些条件时跑死循环，从而让测评返回超时，我们称之为欺骗性提交。
- if(a > 100)while(1);
- 非常简单粗暴，如果运行中有个变量满足某个条件，就进行死循环。
- ACM 中由于错误提交有罚时，所以欺骗提交其实很昂贵，不到紧要关头不会使用。
- 但 PAT 优先考虑得分，并不非常重视提交次数，所以有时也是可行手段。

例题 B1034 有理数四则运算

```
int d1 = 1, d2 = 1;
int a1, b1, a2, b2;
scanf("%d/%d%d/%d",&a1,&b1,&a2,&b2);
if(a1 < 0)d1 = -1, a1 = -a1;
if(a2 < 0)d2 = -1, a2 = -a2;
if(b1 >= 100000 || b2 >= 100000
        || a1 >= 100000 || a2 >= 100000)while(1);
```

时间	结果	得分	题目
9月07日 00:53	部分正确	14	1034

测试点

测试点	结果	用时(ms)
0	答案正确	0
1	答案正确	0
2	运行超时	
3	运行超时	

查看代码

例题 B1034 有理数四则运算

- 有两组数据超过了 1e5
- 我们之前讲过 1e5*1e5 是超过 int 范围的
- 因此很有可能就是因为这个出了错

数值范围

- char : [-127 , 128]
- 不要用 char 作为计数器
- int : - 2e9 ~ 2e9
- 1e5 × 1e5 就会超过 int
- long long： -9e18~9e18
- 满足各种 int 运算需要，但是不适用于大数（几百位的数）
- long long a; scanf(“%lld”,&a); printf(“%lld”,a);

例题 B1034 有理数四则运算

- 我们将中间运算时的变量定义为 long long 类型，就能避免数值溢出的问题
- 但是 __gcd 并没有对 long long 的重载，因此我们需要手写 gcd 函数
- 并且需要注意将两个 int 相乘过程中也需要强制类型转换成 long long 来进行运算，否则在运算过程中就会溢出

例题 B1034 有理数四则运算

- 我们将中间运算时的变量定义为 long long 类型，就能避免数值溢出的问题
- 但是 __gcd 并没有对 long long 的重载，因此我们需要手写 gcd 函数
- 并且需要注意将两个 int 相乘过程中也需要强制类型转换成 long long 来进行运算，否则在运算过程中就会溢出
- 1LL*a*b
- a*(long long)b

例题 B1020 月饼

- 月饼是中国人在中秋佳节时吃的一种传统食品，不同地区有许多不同风味的月饼。现给定所有种类月饼的库存量、总售价、以及市场的最大需求量，请你计算可以获得的最大收益是多少。
- 注意：销售时允许取出一部分库存。样例给出的情形是这样的：假如我们有 3 种月饼，其库存量分别为 18、 15、 10 万吨，总售价分别为 75、 72、 45 亿元。如果市场的最大需求量只有 20 万吨，那么我们最大收益策略应该是卖出全部 15 万吨第 2 种月饼、以及 5 万吨第 3 种月饼，获得 72 + 45/2 = 94.5 （亿元）。

例题 B1020 月饼

- 输入格式:
- 每个输入包含 1 个测试用例。每个测试用例先给出一个不超过 1000 的正整数 N 表示月饼的种类数、以及不超过 500 （以万吨为单位）的正整数 D 表示市场最大需求量。随后一行给出 N 个正数表示每种月饼的库存量（以万吨为单位）；最后一行给出 N 个正数表示每种月饼的总售价（以亿元为单位）。数字间以空格分隔。
- 输出格式:
- 对每组测试用例，在一行中输出最大收益，以亿元为单位并精确到小数点后 2 位。

例题 B1020 月饼

- 输入样例:
- 3 20
- 18 15 10
- 75 72 45
- 输出样例:
- 94.50

例题 B1020 月饼

- 思路：很明显，由于题目中描述了，允许取出一部分库存销售，因此按照常理思考，要想获得最大收益，当然是尽量卖贵的，也就是单价高的。
- 而且我之前也说过， PAT B 组最后一题，一般都是排序之后找思路。
- 所以很容易想到，我们可以将不同的月饼按照其单价排序，然后看能卖多少就卖多少。

例题 B1020 月饼

- 思路：很明显，由于题目中描述了，允许取出一部分库存销售，因此按照常理思考，要想获得最大收益，当然是尽量卖贵的，也就是单价高的。
- 而且我之前也说过， PAT B 组最后一题，一般都是排序之后找思路。
- 所以很容易想到，我们可以将不同的月饼按照其单价排序，然后看能卖多少就卖多少。
- 那么问题就是怎么方便对两个值排序了。
- 定义 struct 并对其排序！

sort 对于 struct 的定制

- 有时我们需要一个 struct 类型进行排序，可是 struct 并没有自然的比较大小，那么我们需要 sort 的时候怎么写重载?

sort 对于 struct 的定制

- 一种是重载 struct 的小于号，也就是给这个结构体定义一个小于号，这样就能比较大小，也就可以调用 sort 了

```
struct node{
    int a,b;
    bool operator < (const node x)const{
        return a < x.a;
    }
}arr[105];
```

sort(arr, arr + n); 或 sort(arr + 1, arr + n + 1);

sort 对于 struct 的定制

- 另一种是额外写一个比较函数，也可以调用 sort

```
struct node{
    int a,b;
}arr[105];
bool cmp(node x,node y){
    return x.a < y.a;
}

sort(arr, arr + n, cmp); 或
sort(arr + 1, arr + n + 1, cmp);
```

例题 B1020 月饼

- 因此我们可以定义一个月饼的 struct，里面有两个值：数量和总价，并根据这两个值来比较大小

例题 B1020 月饼

- 因此我们可以定义一个月饼的 struct，里面有两个值：数量和总价，并根据这两个值来比较大小

```
struct node{
    int num,sum;
    bool operator < ( const node a)const{
        return sum * 1.0 * a.num > num * 1.0 * a.sum;
    }
}a[maxn];
```

```
sort(a + 1, a + n + 1);
```

例题 B1020 月饼

- 那么接下来，我们就可以根据已经有序的 struct 数组，依次取月饼卖出去了。

例题 B1020 月饼

- 那么接下来，我们就可以根据已经有序的 struct 数组，依次取月饼卖出去了。
- 具体实现是，依次对每种月饼的数量与剩余要卖的指标进行比较，如果数量比指标少，那么就全部卖出，否则卖出等于剩余指标的量的月饼。

例题 B1020 月饼

```
double ans = 0;
for(int i = 1 ; i <= n ; ++i){
    if(d > a[i].num){
        ans += a[i].sum;
        d -= a[i].num;
    }
    else{
        ans += a[i].sum * 1.0 * d / a[i].num;
        break;
    }
}
printf("%.2lf\n",ans);
```

例题 B1020 月饼

- 但是如果仅是这样做，结果是有一组答案错误
- 毫无疑问，思路并没有出现错误，在可以部分售出的情况下要获取最大利润，肯定是按照单价最高的卖这种贪心策略
- 那么问题一定是一些小错误

例题 B1020 月饼

- 再仔细读题，我们发现了一个细节：
- 每个输入包含 1 个测试用例。每个测试用例先给出一个不超过 1000 的正整数 N 表示月饼的种类数、以及不超过 500 （以万吨为单位）的正整数 D 表示市场最大需求量。随后一行给出 N 个正数表示每种月饼的库存量（以万吨为单位）；最后一行给出 N 个正数表示每种月饼的总售价（以亿元为单位）。数字间以空格分隔。

例题 B1020 月饼

- 数量和总价都是正数而不是规定正整数，也就表明可能有浮点数
- 在这样的情况下，我们将读入的 int 改为 double
- 这题就完美通过了
-
- 读题依旧是非常重要的

memset 的注意点

- memset(arr, 0, sizeof(arr));
- memset(arr, -1, sizeof(arr));
- memset(arr, 0x3f, sizeof(arr));

- 特别需要注意，有时数组中的值本身可能有 0，所以 memset 初始化成 0 可能就是不合理的，可能需要初始化成 -1
- 例如 B1065 单身狗

谢谢！