

A 一步直达的车

时间限制: 1000 ms 内存限制: 65536 kb

总通过人数: 1735 总提交人数: 1777

题目描述

象棋中，车走一步可以抵达同一行或同一列的某个位置，给你三个车的坐标，判断有几辆车可以一步到达指定坐标。

输入

第一行输入两个整数 x, y （中间用空格隔开），代表需要抵达的坐标。

第二行输入 $x_1, y_1, x_2, y_2, x_3, y_3$ （中间用空格隔开），代表三辆车的坐标。数据保证坐标合法（即你不用考虑输入不合法的情况，无须进行不合法数据的判断和处理），且车不会阻碍其他车到达指定的坐标。

输出

输出可以一步抵达指定坐标的车的数量。

输入样例

```
1 1
1 7 8 1 3 3
```

输出样例

```
2
```

考察知识点：表达式的值

解题思路：车一步抵达的 `bool` 表达式就是 $(x_i == x || y_i == y)$ ，累加三个表达式的值，得到的就是可以一步抵达指定坐标的车的数量。

代码:

```
#include<stdio.h>

int main() {
    int x, y;
    scanf("%d%d", &x, &y);
    int x1, y1, x2, y2, x3, y3;
    scanf("%d%d%d%d%d%d", &x1, &y1, &x2, &y2, &x3,
&y3);
    printf("%d", (x1 == x | y1 == y) + (x2 == x |
y2 == y) + (x3 == x | y3 == y));
    return 0;
}
```

B 傻傻 Aqi 的聪明计算器

时间限制: 1000ms 内存限制: 65536kb

通过率: 1725/1773 (97.29%) 正确率: 1725/2943 (58.61%)

题目描述

傻傻 Aqi 的数学很不好, 连四则运算都费劲, 就更别说算平方了。于是聪明的 Alice 决定给他用 C 语言编写一个简单计算器, 可是她最近太忙了, 同样聪明的同学们, 你们能帮 Alice 帮 Aqi 编写这个计算器吗? (好多个“帮”, 好拗口啊喂...)

简单计算器要求: 能进行加、减、乘、除运算。

输入

第一项为运算符 `op`，比如：`+`，`-`，`*`，`/`。 第二项为第一个双精度浮点型运算数 `x`，第三项为第二个双精度浮点型运算数 `y`。
保证分母不为 0。

输出

输出运算的结果，即 `x op y` 的值。（结果保留两位小数）

输入样例

```
+
10.19
12.3
```

输出样例

```
22.49
```

HINT

上课认真听讲或者课后认真复习过老师 PPT 的孩子们都应该会呀~

解题思路

解决本题的关键是根据输入的不同的 `op`，选择不同的运算，这就用到了我们刚学过的多分支选择的 `switch` 语句，其一般形式为：

```
switch(表达式){
    case 常量表达式 1: 语句 1;
    case 常量表达式 2: 语句 2;
    ...
    case 常量表达式 n: 语句 n;
    default: 语句 n+1;
}
```

这样的话，就能用该语句根据不同的 `op` 跳转到不同的分支啦。

代码

```
#include <stdio.h>
```

```
int main(){
    char op;
    double x,y,result=0;
    scanf("%c",&op);
    scanf("%lf",&x);
    scanf("%lf",&y);
    switch(op){
        case '+': result=x+y; break;
        case '-': result=x-y; break;
        case '*': result=x*y; break;
        case '/': result=x/y; break;
    }
    printf("%.2f",result);

    return 0;
}
```

PS.

为了保险起见，我们可以将 `case '/': result=x/y; break;` 语句改为 `case '/': if(y!=0){result=x/y;} break;` 来避免分母等于 0 的情况。

值得注意的是，这是在一个 switch 控制结构里嵌套了一个 if 结构，这说明控制结构是允许嵌套的。

C 阿狄的包装纸

时间限制: 1000 ms 内存限制: 65536 kb
总通过人数: 1598 总提交人数: 1708

题目描述

阿狄需要去买一批包装纸来装饰自己准备的礼物，商店里有三种不同类型的，每种包装纸一份的张数和价格都不同，商店不允许将包装拆分购买，为了统一形式阿狄决定只买一种包装纸。现在阿狄想知道购买 n 张包装纸至少需要多少钱？

输入

第一行为需要的包装纸张数 n 接下来三行，每行两个整数，分别为这类包装纸一份的数量和价格

输出

输出一行最少的价钱

输入样例

```
57
50 30
2 2
30 27
```

输出样例

```
54
```

HINT

如果买第一种包装纸，一份只有 50 张，则需要买两份，也就是 $30 \times 2 = 60$ 元

因为买的必须以一份为最小单位，所以注意不能选择单价最低的购买方案，对需要的份数应该向上取整，这里可以用相关的函数 `ceil` 也可以自己 `if` 特判一下

代码：

```
#include<stdio.h>
int main()
{
    int anw,money,num1,price1,num2,price2,num3,price3,numOfAll;
    scanf("%d",&numOfAll);
    scanf("%d%d",&num1,&price1);
    scanf("%d%d",&num2,&price2);
    scanf("%d%d",&num3,&price3);
    money = numOfAll/num1*price1;
    if(numOfAll%num1!=0) money += price1;
    anw = money;
    money = numOfAll/num2*price2;
    if(numOfAll%num2!=0) money += price2;
    if(anw>money) anw = money;
    money = numOfAll/num3*price3;
    if(numOfAll%num3!=0) money += price3;
```

```
if(anw>money) anw = money;
printf("%d",anw);
return 0;
}
```

D 寻找第二小数

时间限制：1000ms 内存限制：65536kb

通过率：1710/1755 (97.44%) 正确率：1710/2795 (61.18%)

题目描述

□你三个整数 a, b, c, □你找出其中第二小的数。

输入

输入包含三个整数 a, b, c, □目保□数据在 int 范□内。

输出

输出第二小数的值。

输入样例

```
1 2 3
```

输出样例

```
2
```

考察内容，判断语句。

思路：对三个数两两进行三次比较，就可以得到一个按升序（或降序）的数列，输出中间元素，即可得到结果。

```
#include <stdio.h>
```

```
int main() {  
    int a, b, c;  
    int d;  
    scanf("%d %d %d", &a, &b, &c);  
    if (a < b) {  
        d = a;  
        a = b;  
        b = d;  
    }  
    if (a < c) {  
        d = a;  
        a = c;  
        c = d;  
    }  
    if (b < c) {  
        d = b;  
        b = c;  
        c = d;  
    }  
}
```

```
printf("%d\n", b);  
  
return 0;  
  
}
```

E 预告片效应

时间限制：1000ms 内存限制：65536kb

通过率：1507/1597 (94.36%) 正确率：1507/3992 (37.75%)

题目描述

忙碌的助教这周看了复联 3 的预告片，充满期待的同时，决定无论如何要抽时间去把《黑豹》看了。于是他打开某个购票 APP，开始挑选一家电影院前去观看。挑选的过程中他发现某些场次与他的课程冲突了，请你求出他能去看的场次。

需要注意的是由于沙河地处偏僻，到达电影院需要 40 分钟，如果到达时间与影片开始时间相同，当作赶上电影，若晚于开始时间，则当作没赶上。

输入

第一行一个时间，格式为小时:分钟，代表助教空闲时间开始的时间点。为了简化输入与输出，整点为 15:0，若小时或分钟只有一位，不需要加 0 补齐，如 7:8 代表七点过八分，第二行输入三个时间，分别为影片开始时间（24 小时制），格式与第一行相同，如 15:40，每个时间以空格隔开。
保证输入的时间均合法。

输出

输出所有可以看的场次开始时间，每个时间一行。如果没有一场可以赶上则输出“NO!”。

输入样例

```
15:0  
15:20 15:45 16:0
```



```
16:0
15:50 16:20 16:35
```

输出样例

```
15:45
16:0

NO!
```

此题主要考察判断语句的使用，难度并不高。

结题思路：

- 1、可以按小时与分钟分别来保存数据，然后在判断预计到达时间与场次开始时间关系时，分别针对小时与分钟判断。需要注意的是，到达时间计算时，需要考虑小时的加一。
- 2、可以将小时换算成分钟，然后只需比较换算后结果即可。

AC 代码：

这里给出第一种方法的代码：

```
#include<stdio.h>
int main(){
    int a1,b1,c1,d1,a2,b2,c2,d2,count=0;
    \\这里引入了一个临时变量用于判断是否三个场次都不能赶上
    scanf("%d:%d%d:%d%d:%d%d:%d", &a1,&a2, &b1,&b2,&c1,&c2,&d1,&d2);
    if(a2+40>60){                //考虑加上 40 分钟后可能小时需要加一
        a1++;
        a2-=20;
    }
    else{
        a2+=40;
    }
    if(a1<b1)                    //针对小时进行大小判断
        printf("%d:%d\n",b1,b2);
    else if(a1==b1){            //若相等再判断分钟的大小关系
        if(a2<=b2)
            printf("%d:%d\n",b1,b2);
        else
            count++;
    }
    else
```

```

        count++;
    if(a1<c1)
        printf("%d:%d\n",c1,c2);
    else if(a1==c1){
        if(a2<=c2)
            printf("%d:%d\n",c1,c2);
        else
            count++;
    }
    else
        count++;
    if(a1<d1)
        printf("%d:%d\n",d1,d2);
    else if(a1==d1){
        if(a2<=d2)
            printf("%d:%d\n",d1,d2);
        else
            count++;
    }
    else
        count++;
    if(count==3)                //如果 count 为三，意味着三场都无法赶上
        printf("NO!");
    count=0;
}

```

F Tarpe 酋长的阿波罗计划

时间限制：1000ms 内存限制：65536kb

通过率：1365/1574 (86.72%) 正确率：1365/8601 (15.87%)

题目描述

部落的王是一条美丽的人鱼，在和朋友交易后，Tarpe 酋长终于获得足够的资金去解救被绑架的王，而现在王在距离为 a 光年的地球，只有达到 0.8 的光速才能及时抵达地球解救王。于是 Tarpe 酋长决定执行阿波罗计划，在这次计划中，连续抛掷三次核弹来对飞船加速，每次核弹爆炸会产生一个极大的瞬时加速度。而 Tarpe 酋长的科技树只能保证飞船依次经受从小到大的冲击，否则就会被核弹炸毁。

现在 Tarpe 酋长希望你能帮他安排一下三次核弹抛掷的顺序，并判断飞船能否及时赶到地球。

输入

多组数据，每组数据一行：

每行四个小数 a, b, c, d (保证都为正值)， a 为部落到地球的距离， b, c, d 分别表示三枚核弹造成的速度改变量

输出

对于每组数据，如果能够及时赶到地球，输出两行，
第一行为核弹顺序，以一个空格隔开，
第二行是到达地球所需要的时间（所有数据保留一位小数），
如果不能，直接输出一行“Out of reach”。

输入样例

```
0.8 0.1 0.5 0.3
1.3 0.3 0.2 0.1
```

输出样例

```
0.1 0.3 0.5
0.9
Out of reach
```

出题背景：

出这道题一方面为了巩固本周所学判断语句的知识，另一方面为了让大家熟悉多组数据输入的形式，为以后的题目预热。这道题真正想考察的是排序的思想，虽然同学们目前所学的知识并不足以解决真正的排序问题，但是三个数的排序完全可以通过 if 语句来实现。

考察知识点：

1. 多组数据输入输出
2. 判断条件控制语句
3. 读题和审题

解题思路：

题干辣么长，你想看吗？（酋长本身也不打算让你们都看）我们必须快速找到重点信息
距离 a ，速度限制条件 0.8，速度改变量 b, c, d ，目标 $b+c+d \geq 0.8$ 并对 b, c, d 排序

酋长挖的坑：

1. 达到意味大于等于，很多同学忽略了等于的情况
2. 输出“out of reach”就是打印“out of reach”吗？那么为什么要说“输出一行”呢？换行符必不可少
3. 每个数据输出之间空一格
4. 排序就是简单的大于或者小于吗？等于的情况是不是也要考虑一下呢？
5. 多组输入就是值不固定的组数，单次输入是没有用的。
6. 通过了样例不代表程序就已经天衣无缝了，酋长精心设计的边界数据是不可能轻易放在样例中的，写程序一定要把问题的特殊情况考虑到。

AC 代码：

```
#include<stdio.h>

int main()
{
    double a,b,c,d; //输入数据可能是小数，double 类型
    while(~scanf("%lf%lf%lf%lf",&a,&b,&c,&d))
    {
        double velocity = b + c + d;    //首先计算速度
        if(velocity<0.8)                  //速度无法达到，直接输出“Out of reach\n”
        {
            printf("Out of reach\n");    //换行符
        }
        else
        {
            double x1,x2,x3;              //准备三个变量，
            if(b>c)                        //先假设一种情况，比如 b>c
            {
                if(b>d)                    //如果 b>d,确定 b 为最大值
                {
                    x3 = b;
                    if(d>c)                //确定中间两个值
                    {
                        x2 = d;
                        x1 = c;
                    }
                }
                else
                {
                    x2 = c;
                    x1 = d;
                }
            }
            else
            {
```

```

        x3 = d;
        x2 = b;
        x1 = c;
    }
}
else
{
    if(c>d)
    {
        x3 = c;
        if(d>b)
        {
            x2 = d;
            x1 = b;
        }
        else
        {
            x2 = b;
            x1 = d;
        }
    }
    else
    {
        x3 = d;
        x2 = c;
        x1 = b;
    }
}
printf("%.1f %.1f %.1f\n",x1,x2,x3);    //打印结果 注意空格和换行
printf("%.1f\n",a/velocity);
}
}
}

```

G Terry 的水仙花数

时间限制：1000ms 内存限制：65536kb

通过率：760/815 (93.25%) 正确率：760/1654 (45.95%)

题目描述

水仙花数（Narcissistic number）也被称为超完全数字不变数（pluperfect digital invariant, PPDi）、自恋数、自幂数、阿姆斯壮数或阿姆斯特朗数（Armstrong number），水仙花数是指一个 n 位数（ $n \geq 3$ ），它的每个位上的数字的 n 次幂之和等于它本身（例如： $13+53+33=153$ ）。现给你两个三位数 a, b ，请你输出 $[a, b]$ 间所有的水仙花数(输入的 a, b 保证有内容输出)。

输入

一行两个数字 a, b 。($100 \leq a \leq b \leq 999$)

输出

由小到大输出 $[a, b]$ 间所有的水仙花数，以空格隔开

输入样例

```
153 153
```

输出样例

```
153
```

HINT

三位的水仙花数个数是有限的(废话)

思路分析

三位的水仙花数只有 4 个，分别是 153、370、371、407。

所以只要检索这 4 个数哪个出现在了区间[a, b]就把它输出即可。

(大家可以放心，在期末上机中不会出现这样考察信息检索能力的题目)

参考代码

```
#include <stdio.h>

int main() {
    int a,b;
    scanf("%d%d",&a,&b);
    if(a<=153&&b>=153) printf("153 ");
    if(a<=370&&b>=370) printf("370 ");
    if(a<=371&&b>=371) printf("371 ");
    if(a<=407&&b>=407) printf("407 ");

    return 0;
}
```

H 忽略前导零

时间限制: 1000 ms 内存限制: 65536 kb

总通过人数: 1621 总提交人数: 1675

题目描述

输入一个五位的数，请忽略前导零后输出。

输入

一个五位的数，可能含前导零。(注意 00000 的情况)

输出

忽略前导零后的数。

输入样例

01024

输出样例

1024

解题思路：

方法一： 根据位数逐渐判断，从第一个不为 0 的位输出或者输出 0。

代码：

```
#include<stdio.h>
```

```
int main() {
```

```
    char c1, c2, c3, c4, c5;
```

```
    scanf("%c%c%c%c%c", &c1, &c2, &c3, &c4, &c5);
```

```
    if (c1 != '0')
```

```
        printf("%c%c%c%c%c", c1, c2, c3, c4, c5);
```

```
    else if (c2 != '0')
```

```
        printf("%c%c%c%c", c2, c3, c4, c5);
```

```
    else if (c3 != '0')
```

```
        printf("%c%c%c", c3, c4, c5);
```

```
    else if (c4 != '0')
```

```
        printf("%c%c", c4, c5);
```

```
    else
```

```
        printf("%c", c5);
```

```
    return 0;
```

```
}
```


方法二： 如果熟悉整数，可以知道整数自动忽略前导零，直接输出即可。

代码：

```
#include<stdio.h>

int main() {
    int a;
    scanf("%d", &a);
    printf("%d\n", a);
    return 0;
}
```