

2018级 航空航天类第三次练习赛题解

By: Ausar

A 小哥哥的区间

分析

题目大致意思就是，给出一堆区间，求交集和并集。

关键点在于两点：

- 多组（不定长）输入数据如何处理
- 如何获得交集和并集

首先，在我们OJ上出现的所谓**多组数据**指的都是输入**长度不定**的数据。一般来说用以下的形式来读入：

```
while(scanf("%...", ...) != EOF){  
    //进入到这里之后，已经完成数据读入了  
    //只需要开始处理数据就好  
}
```

顺便说一下，有的题目需要一边读入一边输出，然后本地测试的时候会出现输入输出交织的情况，这种情况是**正常**的。

我们可以发现，所谓**求交集**，也就是**缩小集合**，**求并集**就是**增大集合**。

所以我们可以维护两个区间，分别代表交集和并集，每读入一个新的区间，就看一下要不要缩小交集，或者要不要扩大并集。

输出的时候判断一下，如果并集的左区间大于右区间，则说明是个空集，输出错误信息就好了。

代码

```
#include <stdio.h>  
//By:Ausar  
int max(int a,int b){  
    return a>b?a:b;  
}  
int min(int a,int b){  
    return a<b?a:b;  
}
```

```

int main(){
    int Al,Ar,Bl,Br,l,r;
    scanf("%d %d",&Al,&Ar);
    Bl=Al;Br=Ar;
    while(scanf("%d %d",&l,&r)!=EOF){
        Al=max(Al,l);//我这里用了自己写的求最大值最小值函数
        Ar=min(Ar,r);//如果你们还没学到函数, 可以用if.....else来写这一部分
        Bl=min(Bl,l);
        Br=max(Br,r);
    }
    if(Al<Ar){
        printf("%d %d\n",Al,Ar);
    }else{
        printf("Empty%%0.9f\\?\n");//注意转义字符
    }
    printf("%d %d",Bl,Br);
    return 0;
}

```

B CWD的掌上明珠

分析

这个题目的意思是, 输入一个字符串。不改变其中的非元音字符的位置, 把元音字符位置反转。

(顺便提一下, 元音指的是a, e, i, o, u 这五个)

这道题目, 我们可以用一个变量i, 从0读到n-1, 也就是从前到后遍历字符串。每遇到一个字符, 进行一下判断:

- 如果这个不是元音, 那么直接原样输出
- 如果这个是元音, 那么从字符串后面找到下一个元音进行输出

为了实现第二点, 我们还需要一个变量j, 初始值为n-1, 每次需要输出一个元音时, 就倒着查找。

代码

```

#include <stdio.h>
//By:Ausar
int isvowel(char a){
    return (a=='a' || a=='e' || a=='i' || a=='o' || a=='u');
}
int main(){
    int n,i,j;
    char ch[666];
    scanf("%d %s",&n,ch);//%s能直接输入一个字符串, 用%s的时候, 不需要在前面加上&
    for(i=0,j=n-1;i<n;i++){
        if(isvowel(ch[i])){//这个地方是我自己写的一个函数, 用于判断是否为元音
            //也可以手动替换成一个大if
            while(!isvowel(ch[j])){//持续查找并移动j, 直到j指向的是一个元音
                j--;
            }
        }
    }
}

```

```

        putchar(ch[j--]); //putchar的作用是输出单个char，比printf写得方便
    }else{
        putchar(ch[i]);
    }
}
return 0;
}

```

C 轰炸四角大楼

分析

这题就是求三角形的外心。百度找公式，然后写下来就好了。

注意要用double,不然可能精度不够。

```

#include<stdio.h>
#include <math.h>
//By:dkq

int main() {
    double x1, y1, x2, y2, x3, y3;
    scanf("%lf%lf", &x1, &y1);
    scanf("%lf%lf", &x2, &y2);
    scanf("%lf%lf", &x3, &y3);
    double a = x1 - x2;
    double b = y1 - y2;
    double c = x1 - x3;
    double d = y1 - y3;
    double e = ((x1 * x1 - x2 * x2) - (y2 * y2 - y1 * y1)) / 2;
    double f = ((x1 * x1 - x3 * x3) - (y3 * y3 - y1 * y1)) / 2;
    double det = b * c - a * d;
    if(fabs(det) < 1e-6){
        printf("(0.00,0.00)\n");
    } else {
        double x0 = -(d * e - b * f) / det;
        double y0 = -(a * f - c * e) / det;
        if(fabs(x0 - 0.0) < 1e-6) x0 = +0.0f;
        if(fabs(y0 - 0.0) < 1e-6) y0 = +0.0f;
        printf("(%.2f,%.2f)\n", x0, y0);
    }
    return 0;
}

```

D 疯

分析

这一题是让我们画出一个图形。难点在于如何计算每行的空格数。

比较坑的一个点是，最后一行的符号个数是根据行的单双数进行区分的。所以得特殊处理一下。

如果总行数是偶数，那么每行长度为 x^2+1 ，如果为奇数，那么为 x^2 。

简便点可以写成： $x*x+(x+1)\%2$;

那么就能很方便地写出下面的代码了

代码

```
#include <stdio.h>
//By:Ausar
int main(){
    int x,i,j;
    scanf("%d",&x);
    for(i=1;i<=x;i++){
        for(j=1;j<=(x*x+(x+1)%2-i*i)/2;j++)//判断一下当前应该输出多少个空格
            putchar(' ');
        for(j=1;j<=i*i/2;j++)//先把左边的$给输出了
            putchar('$');
        putchar(i%2?'$':'@');//这里用到了三元运算符，当i%2为1的时候，输出$，否则输出@
        for(j=1;j<=i*i/2;j++)
            putchar('$');
        putchar('\n');
    }
    return 0;
}
```

E 有钱的qsy

分析

这是一道很裸的**高精度加法**。

题目已经明着告诉了我们了，这题的数据会把long long给炸了。

那么怎么做呢？我们可以开int数组，每一位只存一个10以内的数。

就是把数字的每一位分别存到数组中一个元素里。接着做竖式加法，或者说是手算加法，把对应的位相加，然后处理进位。

输入的时候，需要用字符串的形式来输入。因为只有这样才能把这一串数字读进来。

读好之后，面临两个问题：

1. 如何把char类型的数据转换成int。其实，char也是一个数字，他的值代表了这个字符的ascii码。而在ascii码中，'0'~'9'是连续的，所以我们把一个char减去'0'就能得到对应的int了。
2. 读入字符串的时候，左边的数字是存在数组低位的，而在数字中，这是高位。所以我们需要反转字符串。

代码

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

```
//By:Ausar
int main(){
    char stra[233],strb[233],lena,lenb,maxlen,i,j;
    int sum[233]={}; //把数组元素全部初始化为0
    scanf("%s %s",stra,strb); //读入两个字符串
    lena=strlen(stra); //获取两个数字的长度
    lenb=strlen(strb);
    maxlen=lena>lenb?len:lenb; //获取最长的那个数字的长度
    for(i=0;i<lena;i++){
        sum[i]=stra[lena-i-1]-'0'; //反转数字a, 并且从char转换成int
    }
    for(i=0;i<lenb;i++){
        sum[i]+=strb[lenb-i-1]-'0'; //反转数字b, 并且直接加到结果中
    }
    for(i=0;i<maxlen;i++){ //处理进位问题
        sum[i+1]+=sum[i]/10;
        sum[i]%=10;
    }
    if(sum[maxlen]>0){ //如果最高位出现了进位, 那么更新一下结果的长度
        printf("%d",sum[maxlen]);
    }
    while(maxlen--){
        printf("%d",sum[maxlen]); //输出结果
    }
    return 0;
}
```

F 多项式求值

分析

题目的意思是，给一个多项式，然后计算他的值。

这一题有两种计算方式。

第一种是读入一个项，然后就进行运算并加到答案之中。这种方法的好处是比较简单，但是可能造成累计误差。

第二种方法是先化简，在题目中说了多项式的次数只能是0~10的整数，也就是说，式子一定可以化简成这样的形式：

$$k_0x^0+k_1x^1+k_2x^2+.....k_{10}x^{10}$$

所以我们只需要开一个大小为11的数组k，k[0]~k[10]分别代表k0~k10就好了。

代码

```
#include <stdio.h>
#include <math.h>
int main(){
    int n,k[11]={},tmpk,tmp,i;
    double ans=0,x; //推荐使用精度更高的double而不是float
    scanf("%d",&n);
    while(n--){
```

```

        scanf("%d*x^%d",&tmpk,&tmpp);
        k[tmpp]+=tmpk;//存到相应的地方
    }
    scanf("%lf",&x);
    for(i=0;i<=10;i++){
        ans+=k[i]*pow(x,i);
    }
    printf("%.1f",ans);
    return 0;
}

```

G 最大公约数 PLUS

分析

这一题要我们求多个数字的最大公约数。

其实我们只要求前两个数在公约数，然后每读入一个数字，就把之前算出来的公约数和新的这个数字再求一次公约数，最后得到的结果就是我们想要的。

如何快速求两个数的最小公约数？可以百度一下“辗转相除法”

百度上的方法是基于递归的实现，我在这里给出一个基于循环的实现。

代码

```

#include <stdio.h>
int main(){
    int n,gcd,num,tmp;
    scanf("%d %d",&n,&gcd);//这里用了比较讨巧的方法，把第一个数字当作是最开始的公约数
    while(--n){//while(--n)是循环n-1次，因为之前已经把第一个数字给读入了
        scanf("%d",&num);
        while(num!=0){//辗转相除法
            tmp=num;
            num=gcd%num;
            gcd=tmp;
        }
    }
    printf("%d",gcd);
    return 0;
}

```

H 酸奶想成为魔法少女3

分析

这一道题目也就是一道排序题，但是并不是要我们输出排序之后的结果，而是要输出元素原来的位置。

为了实现这个目的，我们可以开两个数组，分别用来存储最开始的序列，还有他们对应的下标。

当排序的时候需要交换两个数字的时候，把两个数组的内容都给交换就好。

对于排序的算法，由于这个数据比较弱。所以我就选择了一个大家比较好理解的**冒泡排序**

冒泡排序的思想大致是这样的：

进行从0~n-2的循环，如果 $a[i] > a[i+1]$ ，那么就交换这两个元素的位置。

进行一次这样的操作之后，会发现最大的元素像是“浮”到了数组最右边一样，所以称为冒泡排序。

一共有n个元素，所以要进行n次这样的操作。

由于在进行第i次操作的时候，最后面的i-1个元素其实已经是有序了的，所以循环的时候可以只循环从0~n-i-1

代码

```
#include <stdio.h>
void swap(int *a,int *b){//这个函数涉及了指针，有兴趣的同学可以提前预习
    (*a)^=(*b);//这里用到了位运算的异或进行元素交换，是一个骚操作，大家可以自己去搜索一下其原理
    (*b)^=(*a);
    (*a)^=(*b);
}
int main(){
    int a[100],b[100],n,i,j;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
        b[i]=i+1;
    }
    for(i=0;i<n;i++){
        for(j=0;j<n-i-1;j++){
            if(a[j]>a[j+1]){
                swap(&a[j],&a[j+1]);//在这里，为了更加直观，我用的是一个函数对两个元素进行交换
                swap(&b[j],&b[j+1]);//大家在自己写的时候，可以用传统的三变量交换法来进行交换
            }
        }
    }
    for(i=0;i<n;i++){
        printf("%d ",b[i]);
    }
    return 0;
}
```