**2014级C++第七次上机解题报告**

**By 施政彤 14211040**

**jhljx转圈圈**

**解题思路：**

**正如题设所述，本题需要用到二维数组。我们所要做的就是将数字以特别的方式放入这个二维数组，再将他们一一提取出来。那么，如何才能像题目那样让数像转圈圈一样排列在数组中呢？我们可以顺着转圈圈的逻辑想下去，所谓转圈圈，就是在输入时碰到边界时自动转弯，而且在转完一圈后，需要将圈的四条边的单边长度减去一格，以实现向内转圈的功能。所以基本思路就是构造一个大循环，在这个循环内有四条判断语句，分别用于判断在第一到第四条边上输入数的时候是否到达边界需要转弯。而这个循环在何时终止呢？由于笔者数学水平比较渣……笔者通过找规律得出了对于一个n边形，一共需要的转弯数为2\*（n-1）次，这样的话每次转弯都将一个计数器counter++（这样算转弯数会多算一次，所以后面while语句内的判断条件改了），循环写为while（counter<2\*n-1）即可。**

**参考代码：**

**#include<iostream>**

**using namespace std;**

**const int MAX=1000+50;**

**long long a[MAX][MAX]={0};**

**int main()**

**{**

**int n;**

**while(cin>>n)**

**{**

**int num=0;**

**int judge=0;**

**int counter=0;**

**int j=0;**

**int k=0;**

**int xiehe=0;**

**while(counter<=2\*n-1) //前文提到的判断条件**

**{**

**if(k==num&&j==num) //四句if判断**

**{**

**counter++;**

**for( ; k<n-num; k++)**

**{**

**a[j][k]=a[j][k-1]+1; cout<<"a["<<j<<"]["<<k<<"]="<<a[j][k]<<endl;**

**}**

**k--;**

**}**

**if(k==n-num-1&&j==num)**

**{**

**j++;**

**counter++;**

**for( ; j<=n-num-1; j++)**

**{**

**a[j][k]=a[j-1][k]+1; cout<<"a["<<j<<"]["<<k<<"]="<<a[j][k]<<endl;**

**}**

**j--;**

**}**

**if(k==n-num-1&&j==n-num-1)**

**{**

**k--;**

**counter++;**

**for( ; k>=num; k--)**

**{**

**a[j][k]=a[j][k+1]+1;**

**cout<<"a["<<j<<"]["<<k<<"]="<<a[j][k]<<endl;**

**}**

**k++;**

**}**

**if(k==num&&j==n-num-1)**

**{**

**j--;**

**counter++;**

**for( ; j>=num+1; j--)**

**{**

**a[j][k]=a[j+1][k]+1;**

**cout<<"a["<<j<<"]["<<k<<"]="<<a[j][k]<<endl;**

**}**

**j++;**

**}**

**judge=1;**

**if(judge==1)**

**{**

**num++;**

**k++;**

**} //这一段是为了在转完一圈后能让下一圈的单边半径减一**

**}**

**//开始激动人心的输出吧！！！**

**for(int heng=0; heng<n; heng++)**

**{**

**for(int zong=0; zong<n; zong++)**

**cout<<a[heng][zong]<<" ";**

**cout<<endl;**

**}**

**for(int xie=0; xie<n; xie++)**

**xiehe+=a[xie][xie];**

**cout<<xiehe<<endl;**

**}**

**}**

**jhljx下楼梯**

**解题思路：**

**本题的样式其实就是长得比较奇怪的杨辉三角。（//（这句话注释掉了）估计没有人像笔者一样看到竖着的1、2、3、4就以为是等差数列叭。。。QAQ）本题需要我们根据输入的“坐标”读取出所对应的数组位置的数，所以基本思路是我们需要先将杨辉三角输入到这个二维数组中去，然后在输入“坐标”后，首先考虑“坐标”是否存在？存在的话读取出对应的位置的数组的数，不存在的话输出-1即可。在输入杨辉三角时，我们需要先将最左边和斜着的两列1输入数组，然后用递推公式a[i][j]=a[i-1][j-1]+a[i-1][j]得到数组中其他数。**

**参考代码：**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,k,x,y,i,p1;**

**while(cin>>n)**

**{**

**p1=1;**

**int i3,i2;**

**int num=2;**

**long long a[101][101]={0};**

**cin>>k;**

**//下面是输入数组**

**for(i=1;i<=n;i++)**

**{**

**a[i][1]=1;**

**a[i][i]=1;**

**}**

**for(i2=3;i2<=n;i2++)**

**{**

**i3=i2;**

**for( ;i3<=n;i3++)**

**{**

**a[i3][num]=a[i3-1][num-1]+a[i3-1][num];**

**}**

**num++;**

**}**

**//下面是读取**

**for( ;k>=1;k--)**

**{**

**cin>>x>>y;**

**if(x<=n&&y<=x)**

**{cout<<a[x][y]<<endl;}**

**else**

**cout<<-1<<endl;**

**}**

**}**

**}**

**jhljx学排序**

**解题思路：**

**考察宋老师上课讲的冒泡排序。计算交换次数只需要在每次执行交换指令的时候，让一个计数器counter++即可。本题比较基础。**

**参考代码：**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,n,h;**

**int a[1010];**

**while(cin>>n)**

**{**

**int counter=0;**

**for(int s=0; s<n; s++)**

**{**

**cin>>a[s];**

**}**

**for(int pass=0; pass<n-1; pass++)**

**{**

**for(i=0; i<n-1-pass; i++)**

**{**

**if(a[i]<a[i+1])**

**{**

**h=a[i];**

**a[i]=a[i+1];**

**a[i+1]=h;**

**counter++;**

**}**

**}**

**}**

**for(i=0; i<=n-1; i++)**

**{**

**cout<<a[i]<<" ";**

**}**

**cout<<endl<<counter<<endl;**

**}**

**}**

**糖果魔女**

**解题思路：**

**笔者感觉本题主要考察的是对数组的灵活运用，以及思维的缜密性（好吧就是程序猿的避坑性）。**

**At first，说说笔者的基本思路：**

**Step1：开三个数组，分别用来存储糖的质量，含糖量和糖质比。**

**Step2:使用冒泡排序法对糖质比数组进行排序，但是排序后会不知道原来的次序，也就没办法和质量数组与含糖量数组一一对应了，怎么办呢？笔者的答案是再开一个数组，专门作为序号数组，在冒泡排序中糖质比数组中两个数交换时，让它们的序号数组也随之交换，这样就可以知道排序后糖质比以前的序号了。Step3:按照排序后的序号顺次将质量和含糖量分别相加，并且若在某一次相加中质量总和超过上限k，则在相加得到的总数内减去这一次加的值。**

**Step4：输出对应的含糖量总和即可。**

**Then,再说说避坑的事。**

**1.就是在声明数组时要使用double，而非int，否则1/3和1/4就没有差别了。2.不要忘了如果魔女发现吃不下她就会放弃这颗糖**

**3.记得要考虑万一糖果魔女在填饱肚子之前已经把糖吃光了的情况。**

**参考代码：**

**#include<iostream>**

**using namespace std;**

**const int MAX=1000+50;**

**double a[MAX]={0},b[MAX]={0},c[MAX]={0};**

**int p[MAX]={0};**

**int main()**

**{**

**int n,k;**

**while(cin>>n>>k)**

**{**

**int judge=0;**

**double sum=0,sumsweet=0;**

**for(int i=0;i<n;i++)**

**{**

**cin>>a[i];**

**}**

**for(int i=0;i<n;i++)**

**{**

**cin>>b[i];**

**}**

**for(int i=0;i<n;i++)**

**{**

**c[i]=a[i]/b[i];**

**p[i]=i;**

**}**

**for(int mp=0;mp<n;mp++)**

**{**

**for(int mpt=0;mpt<n-mp;mpt++)**

**{**

**if(c[mpt]<c[mpt-1])**

**{**

**swap(c[mpt],c[mpt-1]);**

**swap(p[mpt],p[mpt-1]);**

**}**

**}**

**}**

**for(int l=0;l<n;l++)**

**{**

**sum+=a[p[l]];**

**sumsweet+=b[p[l]];**

**if(sum>k)**

**{**

**sum-=a[p[l]];**

**sumsweet-=b[p[l]];**

**cout<<sumsweet<<endl;**

**judge=1;**

**break;**

**}**

**}**

**if(judge==0)**

**cout<<sumsweet<<endl;**

**}**

**}**

**全能者的悖论**

**解题思路：**

**本题和上一题风格十分类似。同样是开三个数组，只不过把魔女的糖果换成了全能者的石头了而已。本题可以沿用上一题开一个专门的序号数组的方法，不过由于题设只需要得出质量价值第二大的石头的序号，所以大可不必如此。笔者的基本思路如下：先开三个数组，分别储存完毕后，对单位质量价值的数组进行排序，找出最大的那个数，令其等于-1（为什么0不可以呢？因为有可能剩下的第二大的也是价值为0呀~），然后重新对此数组进行排序，找出这次的最大值。最后用一个循环来让数组中的值分别与最大值匹配一下，匹配一致的输出序号即可。**

**参考代码：**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n;**

**double maxi;**

**while(cin>>n)**

**{**

**double a[1050]= {0},b[1050]= {0},c[1050]= {0};**

**for(int q=0; q<n; q++)**

**{**

**cin>>a[q];**

**}**

**for(int p=0; p<n; p++)**

**{**

**cin>>b[p];**

**}**

**for(int o=0; o<n; o++)**

**{**

**c[o]=a[o]/b[o];**

**}**

**maxi=-1;**

**for(int u=0; u<n; u++)**

**{**

**if(c[u]>maxi)**

**{**

**maxi=c[u];**

**}**

**}**

**for(int w=0; w<n; w++)**

**{**

**if(c[w]==maxi)**

**c[w]=-1;**

**}**

**maxi=0;**

**for(int r=0; r<n; r++)**

**{**

**if(c[r]>maxi)**

**maxi=c[r];**

**}**

**for(int t=0; t<n; t++)**

**{**

**if(c[t]==maxi)**

**cout<<t+1<<endl;**

**}**

**}**

**}**

**这货不是二分**

**解题思路：**

**最后一题呢基本思路都是从舒印学长那儿学来的，说难也不难（舒印学长使劲强调这是一道签到题……）,但是说不难吧，也不容易做对。正如题目所述，本题不用二分，只要用检索就可以了。而检索序号的方式就是另笔者这次最涨姿势的地方。为了让检索能便捷，我们可以在数组的序号位置上存为输入的数值，而在内容上存上序号，这样当我们检索到它时，它对应的值就是它的序号了。由于代码就是临摹舒印学长的，就不拿出来啦。**