C++第五次上机解题报告

A **jhljx学gcd**

解题思路：首先求最大公约数最高效的方法为欧几里得发明的辗转相除法，两个整数的最大公约数是能够同时整除它们的最大的正整数。辗转相除法基于如下原理：两个整数的最大公约数等于其中较小的数和两数的相除余数的最大公约数。因此应该用递归来写。

设两数为a、b(a>b），求a和b最大公约数(a，b）的步骤如下：用b除a，得a÷b=q......r1(0≤r1）。若r1=0，则（a，b)=b；若r1≠0，则再用r1除b，得b÷r1=q......r2 (0≤r2）.若r2=0，则（a，b)=r1，若r2≠0，则继续用r2除r1，……如此下去，直到能整除为止。其最后一个非零除数即为（a，b）。

有了最大公约数在用两数乘积除以最大公约数即得最小公倍数。

题目中要求n个数字的最大公约数和最小公倍数，我的思路是每输入一个数，判断其与上次求得的最大公约数的最大公约数，判断其与上次求得的最小公倍数的最小公倍数。最终得到的就是这n个数的公共最大公约与最小公倍数。

参考代码：

#include <iostream>

using namespace std;

int gcd(int,int);

int main()

{

int n,temp,temp1,last1,last2;

while(cin>>n)

{

cin>>last1;

last2=last1;

n--;

while(n--)

{

cin>>temp;

temp1=temp;

if(last1<temp)

swap(last1,temp1);

last1=gcd(last1,temp1);

if(last2<temp)

swap(last2,temp);

last2=last2\*temp/gcd(last2,temp);

}

cout<<last1<<' '<<last2<<endl;

}

}

int gcd(int a,int b)

{

if(a%b==0)

return b;

else

return gcd(b,a%b);

}

B **jhljx学素数**

**解题思路：素数就是只能被1和其本身整除的数字，但应注意1不是素数。简单的思路就是每输入一次数据n，让其从2开始求模直到求到这个数开根号（1**个数 n 如果是合数，那么它的所有的因子不超过sqrt(n)**），如果在这些数字中一直没有能整除n的数，则这个数就是素数，否则就是合数。**

**参考代码：**

#include <iostream>

#include <cmath>

using namespace std;

void abc(long long int);

int main()

{

long long int n;

while(cin>>n)

abc(n);

return 0;

}

void abc(long long int n)

{

if (n==1)

cout<<"jhljx is sangxinbingkuang!"<<endl;

else

{

bool flag=0;

for (int i=2;i<=(int)pow(n,0.5);i++)

{

if(n%i==0)

{

flag=1;

break;

}

}

if(flag)

cout<<"jhljx is sangxinbingkuang!"<<endl;

else

cout<<"jhljx is good!"<<endl;

}

}

C **jhljx学下棋**

**解题思路：题目描述的意思就是每个棋子都会攻击其上下左右四个棋子，因此若要保证所有棋子不相互攻击应该间隔放入。当n为偶数时，每行相间隔正好可以放n/2个棋子，每行都如此，但是相邻的行之间都错开放，一共n行，故能放得棋子数目的最大值为n^2/2。**

**当n为奇数时，应该现在第一行放(n+1)/2个棋子，然后第二行就只能放（n-1）/2个棋子了。此后重复一二行的放法。并且这样最后一行也能放（n+1）/2个棋子，故能放的棋子数目的最大值为（n+1）/2\*(n+1)/2+(n-1)/2\*(n-1)/2。**

**特别应注意的一点就是n\*n可能会超过int的范围，如果声明n为int类型的话可能会出错，因为这样n\*n只能表示int范围内的数字，故声明n时应声明为long long int类型的。**

**参考代码：**

#include <iostream>

using namespace std;

int main()

{

long long int n;

while(cin>>n)

{

long long int temp;

if(n%2==0)

temp=n\*n/2;

else

temp=((n+1)/2)\*((n+1)/2)+((n-1)/2)\*((n-1)/2);

cout<<temp<<endl;

}

return 0;

}

**D jhljx的强迫症**

**解题思路：这道题如果用依次判断从n到n\*（m-1）模上m的余数的话结果会超时，因此应该换一种更高效的方法。可以发现，如果n与m的最大公约不是1的话，则n%m的值将不能够遍历0到m-1中的所有数字，因为所有的余数按从小到大的顺序排的话是以n%m的值为步进值增加的。故应判断n,m的最大公约数，如果是1的话则余数能够遍历0到m-1的所有数字，否则不能。**

**参考代码：**

#include <iostream>

using namespace std;

int gcd(int,int);

int main()

{

int n,m;

while(cin>>n>>m)

{

if(n<m)

swap(n,m);

if(gcd(n,m)==1)

cout<<"jhljxshidadoubi"<<endl;

else

cout<<"shuishuowoshidadoubi"<<endl;

}

}

int gcd(int a,int b)

{

if(a%b==0)

return b;

else

return gcd(b,a%b);

}

**E 汉诺塔再度来袭**

**解题思路：这肯定是要用递归写的题。当将编号为n（n>1）的盘子从一个杆（记为x）移到另一个杆（记为y）时，需先将第n-1个盘子从x杆移到三根杆子中的另一根杆子（记为z）上，然后将编号为n的盘从x杆移动到y杆，最后再将编号为n-1的盘子从z移到编号为n-1的盘子上，即移动到y杆上。若n为1则直接移动。**

**关于总共所需的移动次数：记将n个盘子全部从A移动到B上所需次数为f(n),因为将第n个盘子从A移动到C时，需先将n上的全部n-1个盘子从A移动到B，所需次数与将第n-1个盘子从A移动到C上的次数相同为f(n-1),然后将n移动到C上，再将B杆上的n-1个盘子全部移动到C上，所需次数此时又为f(n-1),故存在这样的关系式，f(n)=f(n-1)\*2+1，（n>1）,f(1)=1,经过计算可得f(n)=2^n-1,即将全部n个盘子从A移动到C需要的最少次数为2^n-1。**

**参考代码：**

#include <cstdio>

#include <cmath>

void abc(int,char,char);

int main()

{

int n;

while(~scanf("%d",&n))

{

int counter=(int)pow(2,(double)n)-1;

printf("%d\n",counter);

abc(n,'A','C');

}

return 0;

}

void abc(int n,char x,char y)

{

char z='A'+'B'+'C'-x-y;

if(n==1)

printf("%d %c->%c\n",n,x,y);

else

{

abc(n-1,x,z);

printf("%d %c->%c\n",n,x,y);

abc(n-1,z,y);

}

}

**F jhljx学斐波那契数列(I)**

**解题思路：签到题，水题……递归写，f(1)=f(2)=1,当n>2时，f(n)=f(n-1)+f(n-2)。**

**参考代码：**

#include <iostream>

using namespace std;

int abc(int);

int main()

{

int n;

while(cin>>n)

cout<<abc(n)<<endl;

}

int abc(int n)

{

if(n==1||n==2)

return 1;

else

return abc(n-1)+abc(n-2);

}

G 找不到的孩子们（真心不知道怎么来的名字）

解题思路：首先应分别讨论a,b中有一方为0或者都为0的情况，然后就是a,b相等但都不为0的情况，这几种情况都是特殊情况，答案是固定的。接下来讨论a,b不相等且都不为0的情况。通过列举几组数据可以很容易的发现每一次的数据中都包含一个子数列为等差数列，且公差为a,b中较小的数，首项为a,b中较大的数。这样可以利用等差数列知识将递减数列的项的数目求出来。并且在数列f(n)中若出现某一项比其相邻的两项都要小，则这个数必定已经出现过了，不用计数器增一。直到递减数列递减到最小非负数项，若为0，则结束，否则应以这个数与a,b中较小的数重新重复刚才的过程，直到某次递减的最小项为0为止。

参考代码：

#include <iostream>

using namespace std;

int main()

{

long long int a,b,ans,temp1,temp2,x;

while(cin>>a>>b)

{

if(a==b&&a==0)

ans=1;

else if(a==b&&a!=0)

ans=2;

else if(a+b!=0&&a\*b==0)

ans=2;

else

{

ans=1;

while(a!=0)

{

if(b>a)

{

a=a+b;

b=a-b;

a=a-b;

}

ans+=a/b;

a=a%b;

}

}

cout<<ans<<endl;

}

}

杨佳伟 14211019