题意就是给出3个原子的化学价，然后组成一个分子，要保证这个分子是稳定的，如果你还记得高中化学知识的话这个很容易理解，然后让你求出1-2 2-3 1-3 号原子之间有几条键， 这里我分别用ta tb tc 表示， 用数学的方法表示出来的话就是a = tc + tb; b = ta+tc; c = ta + tb;可能有多种情况，只要输出一种即可。

我们随便找其中一个原子，然后从0开始枚举它到b原子有多少键，根据上面的式子，可以计算出到c原子的键，然后就可以知另外两个原子间的键值，做一次判断即可，无需判断a = tc + tb; b = ta+tc; c = ta + tb;

当a+b+c是奇数的时候，肯定无解

#include <vector>

#include <list>

#include <map>

#include <set>

#include <queue>

#include <stack>

#include <bitset>

#include <algorithm>

#include <functional>

#include <numeric>

#include <utility>

#include <sstream>

#include <iostream>

#include <iomanip>

#include <cstdio>

#include <cmath>

#include <cstdlib>

#include <ctime>

#include <cstring>

#include <limits>

#include <climits>

#include <cstdio>

#define Fori(x) for(int i=0;i<x;i++)

#define Forj(x) for(int j=0;j<x;j++)

#define maxn 107

#define inf 0x3f3f3f3f

#define ONES(x) \_\_builtin\_popcount(x)

#define \_ << " " <<

using namespace std;

typedef long long ll ;

int main()

{

int a,b,c,ta,tb,tc;

cin>>a>>b>>c;

if(((a+b+c)&1))

{

puts("Impossible") ;

return 0;

}

int i;

bool flag=0;

for(i=0;i<=a;i++)

{

tc=a-i;tb=b-i;

if(tc<0 || tb<0 )continue;

if(tb+tc==c)

{

cout<<i<<' '<<tb<<' '<<tc<<endl;

flag=1;

break;

}

}

if(flag==0)

puts("Impossible");

return 0;

}

法二:

Ab之间连的边，在算a的价数的时候算了，在算b的价数的时候也算了，所以就是算了两次

#include <iostream>

#include <stdio.h>

using namespace std;

int main()

{

int a, b, c;

cin>>a>>b>>c;

int sum=a+b+c;

if((sum&1))

{

puts("Impossible");

return 0;

}

sum>>=1;

int x=sum-c,y=sum-a,z=sum-b;

if(x<0 ||y<0 ||z<0)

puts("Impossible");

else cout<<x<<' '<<y<<' '<<z<<endl;

return 0;

}

法三：

设ab之间的边数为x，则bc之间边数为b-x，ac之间边数为a-x

那么c的连边为c=a+b-2x

枚举判断即可

#include <iostream>

#include <stdio.h>

using namespace std;

int main()

{

int a, b, c;

cin>>a>>b>>c;

if((a+b+c)&1)

{

puts("Impossible");

return 0;

}

bool flag=0;

for(int x=0;x<a+b+c;x++)

{

if(a-x>=0 && b-x>=0 && a+b-2\*x==c)

{

cout<<x<<' '<<b-x<<' '<<a-x<<endl;

flag=1;

break;

}

}

if(flag==0)puts("Impossible");

return 0;

}