对于一个长度为n的序列，可以移动n/2次

题目大意：给出两个整数n和k，找出n个不同的数。如果n为偶数，gcd(a1,a2)+gcd(a3,a4)+……+gcd(an-1,an)=k；如果n为奇数，gcd(a1,a2)+gcd(a3,a4)+……+gcd(an-2,an-1)=k。

其实自己差不多想出来了，但是在输出-1的情况下没考虑全面

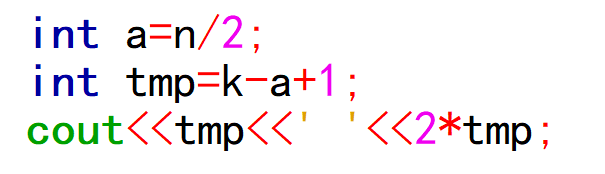
分析：因为每两个数得出一个数，设a=n/2，

如果a>k,即每两个数的最大公约数都为1时都不能满足题意，输出-1；如果k=0,且n>=2时，同样输出-1。

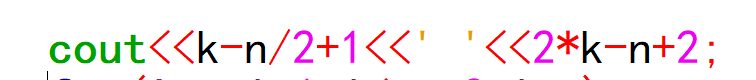
如果a=k,那么只要使每一组的最大公约数为1即可。

如果a<k，令a-1组的最大公约数为1，第一组的最大公约数为k-a+1即可。

**注意**

****

**和**

****

**是不一样的**

**当k=2,n=5的时候就不一样，甚至第二个式子都出来负数了**

#include <cstdio>

#include <cstring>

#include<iostream>

#include<algorithm>

using namespace std;

int main()

{

int n,k;

cin>>n>>k;

int t=n/2;

if(t>k || (t==0 && k!=0))//注意t==0包含了n=0和1的情况

{

puts("-1");

return 0;

}

if(n==2)

{

cout<<k<<' '<<2\*k<<endl;

return 0;

}

if(t==k)

{

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

cout<<i<<(i==n?'\n':' ');

return 0;

}

int tmp=k-t+1;

cout<<tmp<<' '<<2\*tmp;

for(int i=1;i<=n-2;i++)

cout<<' '<<2\*tmp+i;

cout<<endl;

return 0;

}