烧烤架上有n个烤串并排放着，每次选择一个串反转，同时这个串的左k个串和右k个串也会反转，问能实现将所有串都反转的最小次数是多少

枚举，从前k+1个串选择第一个要选择的串，再依次往后找，如果能行，就更新

肯定是不能重叠的，如果重叠了你就万劫不复了，中间是反转过而来的，两边是没反转的，你不可能再通过反转实现把他们全翻转到一个面了

#include<bits/stdc++.h>

#include<cstring>

using namespace std;

#define ll long LONG\_MAX

#define INF 1000000007

int main()

{

//freopen("input.txt","r",stdin);

int n,k;

cin>>n>>k;

// if(k==0)

// {

// cout<<n<<endl;

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

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

// return 0;

// }

// if(n==1)

// {

// cout<<1<<endl<<1<<endl;

// return 0;

// }

// if(n==2\*k+1)

// {

// cout<<1<<endl;

// cout<<k+1<<endl;

// return 0;

// }

// if(2\*k+1>n)

// {

// cout<<1<<endl;

// cout<<n/2<<endl;

// return 0;

// }

// if(n%(2\*k+1)==0)

// {

// cout<<n/(2\*k+1)<<endl;

// for(int i=k+1;i<=n;i+=2\*k+1)

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

// return 0;

// }

int ans=INF;

vector<int>store;

for(int p=1;p<=k+1;++p)

{

vector<int>v;

int now=p;

bool sf=0;

while(now<=n)

{

v.push\_back(now);

if(now+k>=n)

{

sf=1;

break;

}

now+=2\*k+1;

}

if(sf)

if(ans>v.size())

{

ans=v.size();

store=v;

}

}

cout<<ans<<endl;

for(int i=0;i<store.size();++i)

cout<<store[i]<<(i==store.size()-1?'\n':' ');

return 0;

}