题意：一个长度为N(<=5e4)的数组A，从A中选出若干个数，使得这些数的和是N的倍数。  
例如：N = 8，数组A包括：2 5 6 3 18 7 11 19，可以选2 6，因为2 + 6 = 8，是8的倍数。

思路：以前看过抽屉定理，觉得这个定理废话。但是看到这道题，感觉真的好神！

因为只有n个数，如果这n个数中，有其中一个数%n为0，那么肯定是直接输出

如果所有的数%n都不为0，那么就可能为1~n-1里的任何一个，但是有n个数。

这就说明，至少有一个数字，会存在2次！

这样看起来没用，但是如果我是维护前缀和，那就有用了。

如果一个前缀和的值出现了2次，我们都知道，那么这一段区间里的数字之和%n就会等于0，那就是答案了

#include<bits/stdc++.h>

typedef long long ll;

using namespace std;

#define eps 1e-9

ll a[50010],f[50010];

int main()

{

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

int n;

cin>>n;

cin>>a[0];f[0]=a[0]%n;

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

{

cin>>a[i];

f[i]=(f[i-1]+a[i])%n;

}

//for(int i=0;i<n;i++)cout<<f[i]<<' ';cout<<"\n";

bool flag=0;

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

if(f[i]==0)

{

flag=1;

cout<<i+1<<"\n";

for(int j=0;j<=i;j++)

cout<<a[j]<<"\n";

break;

}

if(flag==0)

{

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

if(flag==0)

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

if(f[i]==f[j])

{

cout<<j-i<<"\n";

for(int k=i+1;k<=j;k++)//注意不是从i到j，是从i+1到j，个数也是j-i个，不是j-i+1个

cout<<a[k]<<"\n";//因为前缀和的差得到的就是i+1到j，不包含i

flag=1;

break;

}

}

return 0;

}