

## POJ.1721.CARDS

题意：给定 $n, s$ 和排列 $a_1, a_2 \dots, a_n$ ，若置换 $p^s = a$ ，求置换 $p$ 。

考虑先找到排列 $a$ 的循环节 $len$ ，即 $a$ 置换 $len - s \% len$ 次能得到 $p$ ，因为 $[(len - s \% len) + s] \% len = 0$ ，即为 $a$ ，所以暴力操作特判即可。

```
#include<cstdio>
#include<iostream>
#include<cstring>
using namespace std;
typedef long long ll;
const int N=1e3+5,M=2e4+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
#define pb push_back
int a[N],b[N],tmp[N],n,s;
void change(){
    for(int i=1;i<=n;i++) tmp[i]=b[b[i]];
    for(int i=1;i<=n;i++) b[i]=tmp[i];
}
bool check(){
    for(int i=1;i<=n;i++) if(a[i]!=b[i]) return 0;
    return 1;
}
int main(){
    scanf("%d%d",&n,&s);
    for(int i=1;i<=n;i++) scanf("%d",&a[i]),b[i]=a[i];
    int len=1;
    for(;;len++){
        change();
        if(check()) break;
    }
    int cnt=len-s%len;
    for(int i=1;i<=cnt;i++)
        change();
    for(int i=1;i<=n;i++)
        printf("%d\n",b[i]);
    return 0;
}
```

## 牛客多校2.Just Shuffle

题意：给定排列 $a_1, a_2, \dots, a_n$ ，若 $p^k = a$ ，求置换 $p$ 。

因为： $p^k = a \rightarrow p^{kt} = a^t$ ，当 $t$ 为 $k$ 的逆元时，有 $p = a^t$ ，因此我们只要求出 $a^t$ 即可。 $t$ 就是 $k$ 的逆元，我们依次求出每个循环节的长度 $len$ ，然后暴力找逆元 $inv \times k \% len = 1$ ，

根据公式： $ans[h[i]] = h[(i + inv) \% len]$ 即可得到 $p$ 。

$h[i]$ 代表循环节中第 $i$ 个数。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+5,M=2e4+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
#define pb push_back
int n,k,a[N],vis[N],ans[N];
vector<int>h;
void fun(){
    int len=h.size(),inv;
    for(int i=0;i<len;i++)
        if(1LL*i*k%len==1) inv=i;
    for(int i=0;i<len;i++){
        printf("h[%d]=%d\n",i,h[i]);
        ans[h[i]]=h[(i+inv)%len];
    }
}
int main(){
    scanf("%d%d",&n,&k);
    for(int i=1;i<=n;i++) scanf("%d",&a[i]);
    for(int i=1;i<=n;i++){
        int x=a[i];
        if(!vis[x]){
            h.clear();
            while(!vis[x]){
                vis[x]=1;
                h.push_back(x);
                x=a[x];
            }
            fun();
        }
    }
    for(int i=1;i<=n;i++)
        printf("%d ",ans[i]);
    return 0;
}
```