

## E.Putting Candies

用 $\text{candy}[i]$ 表示第 $i$ 次操作后的糖果个数, $A[i]$ 表示第 $i$ 个盘子的糖果个数  
有:  $\text{candy}[i+1] = \text{candy}[i] + A[\text{candy}[i] \% N]$   
记录  $\text{candy}[i] \% N$  的出现位置,当第二次出现时就找到了一个循环  
然后计算即可.

```
using ll = int64_t;

const int max_n = 2e5+10;
int N,A[max_n],pre[max_n];
ll candy[max_n];
int p,st = 0,ed;
ll a,b,p_candy;
ll ans = 0,K;

void solve(){
    for(int i = 0;i <= N-1;i++){
        candy[i+1] = candy[i] + A[candy[i] % N];
        if(pre[candy[i+1] % N] != -1){
            st = pre[candy[i+1] % N];
            ed = i+1;
            break;
        }
        pre[candy[i+1] % N] = i+1;
    }
}

int main(){
    scanf("%d %lld",&N,&K);
    for(int i = 0;i < N;i++)
        scanf("%d",&A[i]);
    for(int i = 0;i < N;i++)
        pre[i] = -1;

    candy[0] = 0;
    pre[0] = 0;
    solve();

    if(K <= st)
        ans = candy[K];
    else{
        p = ed - st;
        p_candy = candy[ed] - candy[st];
        a = (K-st-1)/p;
```

```
        b = (K-st-1)%p;
        ans = candy[st+b+1] + (a*p_candy);
    }
    printf("%lld\n",ans);
    return 0;
}
```