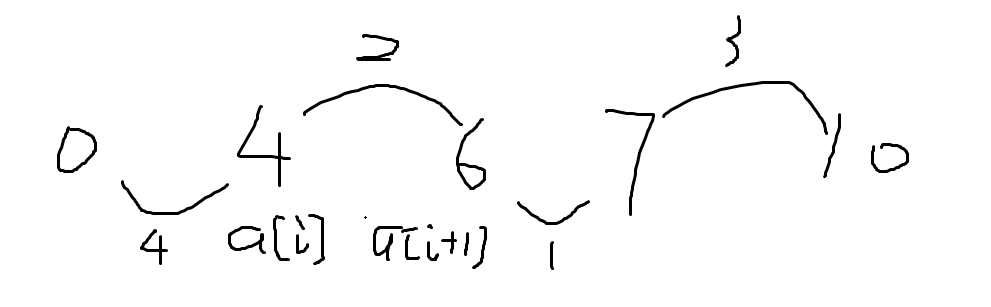
比如第一个样例：开关灯时长为：4(开) 2(关) 1(开) 3(关) ，我们加一次操作相当于把某个数字拆成两个，肯定为一开一关，我们肯定让开最大（即该数字-1），拆完之后的开灯时长为：该数字之前的总开灯时长+该数字之后的总关灯时长+该数字-1。   
记录到第i次操作时亮灯时长记为up[i]。每次操作后开灯时长为：up[i]+m-a[i+1]-(up[n+1]-up[i+1])+a[i+1]-a[i]-1，所以遍历一遍维护最大值即可。别忘了可以不操作，此时开灯时长为up[n+1]。



例如现在算a[i]-a[i+1]这一段

up[i]是i之前的亮灯时长

m-a[i+1]是a[i+1]到终点的时长

up[n+1]-up[i+1]是i+1到终点的亮灯的时长

所以m-a[i+1]-(up[n+1]-up[i+1])就是i+1到终点的灭灯时长

A[i+1]-a[i]-1就是当前段我们拆分，肯定让亮的最大，就让灭的时长最小，最小是1

不管我当前这的这一段（a[i]~a[i+1]）是开还是关，只要我插进去一个。他就分为两段，那么这一段后面的必然全部反转，亮变灭，灭变亮

#include <bits/stdc++.h>

#define ll long long

using namespace std;

#define INF 1000000007

ll a[100010];

ll up[100010];//up[i]代表1-i开灯时长

int main()

{

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

int n;

ll m;

cin>>n>>m;

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

cin>>a[i];

bool state=1;

a[0]=0;a[n+1]=m;

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

{

up[i]=up[i-1]+state\*(a[i]-a[i-1]);

state=!state;

}

up[n+1]=up[n]+state\*(m-a[n]);

ll ans=up[n+1];

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

if(a[i+1]-a[i]>1)

{

ans=max(ans,up[i]+m-a[i+1]-(up[n+1]-up[i+1])+a[i+1]-a[i]-1);

}

cout<<ans<<endl;

return 0;

}