题意：给你三个数n,a,b,定义s[i]=a^i%b,T i = Min{ S k | i-A <= k <= i, k >= 1} ,要你求ti数组累成起来的积再取余b

思路：如果懂得单调栈的思想应该很容易就能想到这题要用单调栈来解决，模型不难建立，维护s[i]数组的单调递增序列然后每次把栈底元素累成起来取余b即可

首先a[i]=An %B 嘛

那这个怎么求，我一开始做的时候先快速幂把a[1]-a[n]都求了出来，之后再用单调队列来算的，但这样T了

看了别人的做法，因为a[i]=a[i-1]\*A%B, 这样在用单调队列算的过程中求就可以了！！！

还有一点

N是1e7的，所以如果a数组开long long的话内存不够

可以看出%B之后的值是int型的，那就把a开为int型，但注意在算的过程中机会极有可能会爆int，所以要注意转long long

#include<cstdio>

#include<iostream>

#include<vector>

using namespace std;

typedef long long ll;

const int INF=1000000007;

int a[10000010];

int q[10000010];

int main()

{

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

int n,A,B;

while(~scanf("%d%d%d",&n,&A,&B))

{

int l=1,r=0;

ll ans(1);

a[0]=1;

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

{

a[i]=a[i-1]\*(ll)A%B;

while(i-A>q[l])

l++;

while(l<=r && a[q[r]]>a[i])

r--;

q[++r]=i;

ans=(ans%B\*a[q[l]]%B)%B;

}

printf("%lld\n",ans);

}

return 0;

}