先前接触了单调队列的题，确实是很重要的知识点，单调队列不难，就是用栈或队列来通过对队头的判断来实现队列保持单调。

题意：

给出一个大小为n的数组a[n];

求其中最大值减最小值在【m,k】中的字串最长长度。

思路：

用两个单调队列分别维护a【i】前元素中的最大值与最小值的下标，top为最值。

然后当最值之差过大时，a【i】的满足题意的最长字串为最最后操作last与【i】的距离  其中last取离i最远的一个

这样就出来了，单调队列主要是在于数组操作上，要将数组各种变量与题意紧紧联系起来。。。

#include<iostream>

#include<stdio.h>

#include<queue>

#include<stdio.h>

using namespace std;

int main()

{

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

int n,m,k;

int a[100010];

while(~scanf("%d%d%d",&n,&m,&k))

{

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

scanf("%d",&a[i]);

deque<int>q1;

deque<int>q2;

int ans(0),last1(-1),last2(-1);//这个last要赋值成第一个元素的前一个坐标

//就是为了防止所有元素大小都相等，last没有更新过这种情况

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

{

while(!q1.empty() && a[q1.back()]<=a[i])

q1.pop\_back();

q1.push\_back(i);

while(!q2.empty() && a[q2.back()]>=a[i])

q2.pop\_back();

q2.push\_back(i);

while(a[q1.front()]-a[q2.front()]>k)

if(q1.front()<q2.front())

{

last1=q1.front();

q1.pop\_front();

}

else

{

last2=q2.front();

q2.pop\_front();

}

if(a[q1.front()]-a[q2.front()]>=m)

ans=max(ans,i-max(last1,last2));

}

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

}

return 0;

}