题目大意：有n个数你能对他们进行k次操作每次操作对其中一个数乘x问如何操作之后使其两两按位或值最大

解题思路:为了使值最大很显然应该操作二进制位数最多的数，但注意不是想当然的对最大ai进行操作就会获得最大值二进制位数同等数量的值具有相同的机会所以应该枚举。但不能直接暴力每次从头到尾|这样会超时，可以使用前缀数组和后缀数组。   
此处的前缀和指某数组的前i项和。   
如给定数组a[n]，求sum[n]。其中sum[i]=a[0]+a[1]+…+a[i]   
这里的sum[]数组即为数组a的前缀和数组。   
那么前缀和有什么用处呢？   
假设我要求a[x]+a[x+1]+a[x+2]+…+a[y]   
那么这个答案就等于 sum[y]-sum[x-1](https://blog.csdn.net/qq_36290842/article/details/x!=0)。若x=0则等于sum[y]。   
有一点：如果原数组的数字是非负的，那么sum数组则是非递减的，在搜索的时候可以考虑使用二分。（后缀同理）

#include <iostream>

#include <cstring>

#include <cstdio>

#include <string>

#include <iomanip>

#include <vector>

#include <map>

#include <set>

#include <algorithm>

#include <queue>

#include <cmath>

using namespace std;

#define ll long long

ll a[200010],sum[200010],beh[200010];

int main()

{

int n,k,x;

scanf("%d%d%d",&n,&k,&x);

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

scanf("%I64d",&a[i]);

ll mul=1;

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

mul\*=x;

sum[1]=a[1];

for(int i=2;i<=n;i++)

sum[i]=sum[i-1]|a[i];

beh[n]=a[n];

for(int i=n-1;i>=1;i--)

beh[i]=beh[i+1]|a[i];

ll ans=0,cnt=0;

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

{

cnt=sum[i-1]|(a[i]\*mul)|beh[i+1];

ans=max(ans,cnt);

}

printf("%I64d\n",ans);

return 0;

}