N个整数组成的循环序列a11,a22,a33,…,ann，求该序列如aii+ai+1i+1+…+ajj的连续的子段和的最大值（循环序列是指n个数围成一个圈，因此需要考虑an−1n−1,ann,a11,a22这样的序列）。当所给的整数均为负数时和为0。

例如：-2,11,-4,13,-5,-2，和最大的子段为：11,-4,13。和为20。

**Input**

第1行：整数序列的长度N（2 <= N <= 50000)   
第2 - N+1行：N个整数 (-10^9 <= Sii <= 10^9)

**Output**

输出循环数组的最大子段和。

**Sample Input**

6

-2

11

-4

13

-5

-2

**Sample Output**

20

**分析：**本题与普通的最大子段和问题不同的是，最大子段和可以是首尾相接的情况，即可以循环。那么这个题目的最大子段和有两种情况

    （1）正常数组中间的某一段和最大。这个可以通过普通的最大子段和问题求出。

    （2）此数组首尾相接的某一段和最大。这种情况是由于数组中间某段和为负值，且绝对值很大导致的，那么我们只需要把中间的和为负值且绝对值最大的这一段序列求出，用总的和减去它就行了。

     即，先对原数组求最大子段和，得到ans1，然后把数组中所有元素符号取反，再求最大子段和，得到ans2，

     原数组的所有元素和为ans，那么最终答案就是**max(ans1, ans + ans2)**。

#include<iostream>

#include<cstring>

using namespace std;

typedef long long ll;

ll a[50010];

int main()

{

int n;

cin>>n;

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

cin>>a[i];

ll sum(0),ans1(0);

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

{

sum+=a[i];

if(sum<0)sum=0;

if(sum>ans1)ans1=sum;

}

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

a[i]=-a[i];

sum=0;

ll ans2=0;

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

{

sum+=a[i];

if(sum<0)sum=0;

if(sum>ans2)ans2=sum;

}

sum=0;

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

sum-=a[i];//之所以是减是因为a[i]取反了

cout<<max(ans1,sum+ans2)<<"\n";

return 0;

}