**MAXIMUM CIRCULAR SUM**

You are provided n numbers (both +ve and -ve). Numbers are arranged in a circular form. You need to find the maximum sum of consecutive numbers.

**Input Format:**

First line contains integer t which is number of test case.  
For each test case, it contains an integer n which is the size of array and next line contains n space separated integers denoting the elements of the array.

**Constraints:**

1<=t<=100  
1<=n<=1000  
|Ai| <= 10000

**Output Format**

Print the maximum circular sum for each testcase in a new line.

**Sample Input**

1

7

8 -8 9 -9 10 -11 12

**Sample Output**

22

Program-

#include<iostream>

using namespace std;

int kadane(int a[],int n)

{

int cs=0;

int ms=0;

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

{

cs=cs+a[i];

if(cs<0)

{

cs=0;

}

ms=max(cs,ms);

}

return ms;

}

int maxCircularSum(int a[],int n)

{

int sum=kadane(a,n);

int csum=0;

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

{

csum=csum+a[i];

a[i]=-a[i];

}

int msum=csum+kadane(a,n);

if(sum>msum){

return sum;

}

else{

return msum;

}

}

int main() {

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int a[n];

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

{

cin>>a[i];

}

cout<<maxCircularSum(a,n)<<endl;

}

}