**Максимална сума от поредни елементи**

Дадено ви е цяло положително число N и масив А с N целочислени елемента. Намерете най-голямата сума на поредни елементи от този масив. (Задачата е давана в миналото на приемен изпит в софийската фирма Mobisystems - автор на най-популярния офис пакет "Office Suite" за OS Android преди излизането на мобилната версия на Microsoft Office)  
Жокер: Наивното решение би било с два вложени цикъла да изчислим сумата навсички подредици на А, но това би ни довело до 1,000,000 ^ 2 = 10^12 елементарни изчисления, което е много повече от позволените 10^9 за да влезем в даденото от журито време за изпълнение на програмата (обикновено 2 секунди).  
Но дали наистина имаме нужда от пробването на всички подредици? Представете си че в момента сте изчислили сумата на подредицата от А[CurrentStartPos] до А[CurrentFinalPos].  
Въпрос 1: Ако тази сума е отрицателна (равна на SomeNegativeNumber) има ли смисъл да смятаме сумите на следващите подредиците с край от А[CurrentFinalPos + 1] до А[N-1]. За да има смисъл, то една от тези подредици, завършваща на А[MaxSumFinalPos], трябва да ни даде максималната сума MaxSum от подредица на А, която търсим. Но ако това е така, то каква сума би ни дала подредицата започваща от А[CurrentFinalPos+1] до А[MaxSumFinalPos] при положение че:  
MaxSum = А[CurrentStartPos] + А[CurrentStartPos + 1] + .. + А[CurrentFinalPos] + А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos]  
Тъй като вече имаме че: А[CurrentStartPos] + А[CurrentStartPos + 1] + .. + А[CurrentFinalPos] = SomeNegativeNumber  
Заместваме последното равенство в предпоследното и получаваме:  
MaxSum = SomeNegativeNumber + А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos]  
=>А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos] = MaxSum - SomeNegativeNumber  
=>А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos] = MaxSum + SomePositiveNumber  
=>А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos] = MoreThanMaxSum  
Следователно достигаме до противоречие че редицата А[CurrentStartPos] до А[CurrentFinalPos + SomeOtherPositiveNumber] би могла да бъде максимална, тъй като редицата А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[CurrentFinalPos + SomeOtherPositiveNumber] би ни дала по-голяма сума от нея. От това на свой ред следва че НЯМА СМИСЪЛ ДА ПРОДЪЛЖАВАМЕ ДА ПРОБВАМЕ ДРУГИ ПО-ДЪЛГИ ПОДРЕДИЦИ ЗАПОЧВАЩИ ОТ А[CurrentStartPos], АКО ВЕЧЕ СМЕ ПОЛУЧИЛИ ОТРИЦАТЕЛНА СУМА НА ТЕКУЩАТА ПОДРЕДИЦА ЗАПОЧВАЩА ОТ А[CurrentStartPos].  
Въпрос 2: Ако подредицата А[CurrentStartPos] до А[CurrentFinalPos] е достигнала вече отрицателна сума CurrentNegativeSum и всички предни подредици със същото начало са имали положителни суми (следствие от отговора на предния въпрос) има ли смисъл следващият CurrentStartPos (който ще обозначим с NewStartPos) да е в интервала CurrentStartPos+1 до CurrentFinalPos? Ако имаше то означава че редица започваща от позиция NewStartPos в интервала от CurrentStartPos+1 до CurrentFinalPos потенциално би била максимална и равна на MaxSum:  
MaxSum = А[NewStartPos] + А[NewStartPos + 1] + .. + А[CurrentFinalPos] + А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos]  
Заместваме А[NewStartPos] + А[NewStartPos + 1] + .. + А[CurrentFinalPos] с CurrentNegativeSum - (А[CurrentStartPos] + А[CurrentStartPos + 1] + .. А[NewStartPos-1])  
=> MaxSum = CurrentNegativeSum - (А[CurrentStartPos] + А[CurrentStartPos + 1] + .. А[NewStartPos-1]) + А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos]  
=>(А[CurrentStartPos] + А[CurrentStartPos + 1] + .. А[NewStartPos-1]) = CurrentNegativeSum + (А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos] - MaxSum)  
но MaxSum >= А[CurrentFinalPos + 1] + А[CurrentFinalPos + 2].. + А[MaxSumFinalPos], тъй като дясната част е една от всичките подредици на А  
=>(А[CurrentStartPos] + А[CurrentStartPos + 1] + .. А[NewStartPos-1]) = CurrentNegativeSum + (SomeNonPositiveNumber)  
=>(А[CurrentStartPos] + А[CurrentStartPos + 1] + .. А[NewStartPos-1]) = SomeOtherNegativeNumber2  
От което достигаме отново до противоречие, тъй като по допускане (А[CurrentStartPos].. А[NewStartPos-1]) - която е една от предните подредици - е с положителна сума.  
Следователно НЯМА НУЖДА ДА ТЕСТВАМЕ ЗАСТЪПВАЩИ СЕ ПОДРЕДИЦИ и следващият нов CurrentStartPos ще бъде CurrentFinalPos + 1, когато сме достигнали до отрицателна сума на подредицата А[CurrentStartPos] .. А[CurrentFinalPos].  
Мисля че алгоритъма е вече ясен: Инициализираме търсената резултатна максимална сума с MAX\_NEGATIVE\_INT и текущата сума на подредица с нула (в началото текущата подредица не съдържа елементи). Итерираме всички елементи А[CurrentFinalPos] на масива А и добавяме текущия към текущата сума на подредица (инициализирана на 0 в началото). Максимизираме търсената резултатна сума с текущата сума на подредица. Ако текущата е отрицателна я правим равна на 0 (така неявно правим новата CurrentStartPos да е следващият итериран елемент - CurrentFinalPos + 1) и цикъла по CurrentFinalPos продължава да се върти. Реално не ни трябва да поддържаме явна променлива за CurrentStartPos.

**Input Format**

На първият ред от входа ще получите броят елементи N на масива А. На следващите N реда ще получите поредните целочислени елементи A[i] на масива А.

**Constraints**

1 <= N <= 1,000,000  
-1000 <= A[i] <= 1000

**Output Format**

Изведете най-голямата сума от поредни елементи (един или повече) на масива А.

**Sample Input 0**

6

-1

4

-2

6

1

-7

**Sample Output 0**

9