	□ Logo	
:C408.1	STUDENT REPORT ARCHOS STUDENT REPORT ARCHOS ARCHO	
Choo	ABOUT STUDENT REPORT OF THE CASE OF THE CA	×09
	CWOSY SPINES OSILIN JURICH CITARY MECHOO JURIL CWOSY STRINGER	
DE	ETAILS TAILS TO THE TOTAL TO THE TOTAL TO THE TOTAL TO THE TOTAL T	,
(222	STUDENT REPORT CASE 1 2 BY ALECASE	20
	MOUNIKA	
AECAOS:	Roll Number 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	- 0
AF		
ΕX	22BI24EC408-T REPART PROPRIED TO THE PART OF THE PART	
√2 [™] Tit	le ECHOS JARLA CHOST SPILARE SELLARE JARCA LINE RECHOS	Ý
8	EQUICIBRIUM CROST 278124ECROST NOST 228124ECROST CROST 228124ECROST 228124ECROST 228124ECROST 228124ECROST	b
80,	The state of the s	
,12AECAO8	Description Description Charles Charle	VE.
)`	for the given an array A of N integers. An equilibrium position is a position where the sum of an integers on its left is equal to the sum.	
228	of all integers on its right in the array A. Print the index of the equilibrium position.	
,08-122E	Note: For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT FOUND" without quotes. The array is 1 indexed.	<u></u>
	The array is 1 indexed	
AECA		,
1812AECA	Input Format:	20
5408.TV	The first line contains an integer denoting N.	-8'
SAS	The first line contains an integer denoting N. The second line contains N space-separated integers denoting the elements of the array A.	2
۷.	Input will be read from the STDIN by the candidate	
2281245	Output Format:	BÍ
V.	Print the index of the equilibrium position. If no index is found, print "NOT FOUND"	/
^	Sample Input	
*ECAOS.T	5 	RE
×	24733	,
228120	Sample Output	
22	· **	38,
,	Source Code: ALEC CASE TO THE AND ADDRESS OF THE ADDR	
	27 AECH STADO MECHOS TOBELLOSS. OBSTADOS MESTOS BASE	349
	22th RECADO 2281 CADE 25 2812AT RECADE 25 26 26 26 26 26 26 26 26 26 26 26 26 26	
	Source Code: ALCAOS. T. A. B. A. L. CAOS. T. A. B.	~3
	OBINET OBINE ON DECK ON DECK OF BEING CARANT	Y,
	Source Code: ALC. ADD. T. A. D.	NO.

```
def find_equilibrium_position(N, A):
       total_sum = sum(A)
       left_sum = 0
       for i in range(N):
           right_sum = total_sum - left_sum - A[i]
           if left_sum == right_sum:
               return i + 1
           left_sum += A[i]
       return "NOT FOUND"
   # Input reading
   N = int(input())
   A = list(map(int, input().split()))
                                                                                                   2281 2281 200 AL
   result = find_equilibrium_position(N, A)
   print(result)
RESULT
 5 / 5 Test Cases Passed | 100 %
```