Dogo N N N N N N N N N N N N N N N N N N	1 6
LINGS COLON TO STUDENT REPORT LINGS COLON	2) F118138
1 to 32 set 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	(5¢01)
DETAILS 305 LUBP SELOTIFE SOURCE AND SOURCE SELOTIFE SELOTIF	JB235 EE011*
DETAILS Name Nam	1813CSE011 KUB13CSE011 KUB13CS
B SHREEDHAR	0
	, 14°
^'	- 1
EXPERIMENT, 255 AT LUB 255 AT LUB 25	, KUBL ESEOT
EXPERIMENT CELL TITLE EQUILIBRIUM Description CELL TITLE EQUILIBR	13c5E011 KUB13c5E011 KUB13c5E01
EQUILIBRIUM LUBY CELOT	SE011.
141181 ESEOT (11873C COLT) 140 COLT (1181 CSEOT)	1365
Description Set 11 Lub 3cst 11	2C5* 145
You are given an array A of N integers. An equilibrium position is a position where the sum of all integers on its left i	s equal to the sum
Note: For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NO quotes.	T FOUND" without
	T FOUND" without
The array is 1 indexed.	
Input Format:	.017
	3056011
The input consists of two lines: The first line contains an integer denoting N. The second line contains N space-separated integers denoting the elements of the array A.	
The second line contains N space-separated integers denoting the elements of the array A.	77, KABU
	57
Output Format: Driet the index of the equilibrium position If an index is found point "NOT FOUND"	60
Print the index of the equilibrium position. If no index is found, print "NOT FOUND"	,8 ² 3 ² C5 ^{EO}
Sample Input	
Sample Input 5	ptalify the
24733	in the state of th
Sample Output	
	NE Page
Source Code: St. Land	Fight. Elfaltist physical states of the stat
Source Code: 3C5t LUBP3C5t LUBP3C5E01 LUBP3C	5° Selection of the sel
1823C3 (17KD) 03C5E0 (KUB23 5E0)11 (1828)	SER FOR
1 Light Carlot 1 Ligh	A STATE OF THE STA
Source Code: 3 ^{C5} ^E KUB ² 3 ^{C5} 3 ^{C5} 5 ^C KUB ² 3 ^{C5} 5 ^C KUB	CENTRY LABOR
Source Code: 3C5E 11 KUB13C5E011 KUB13C5E0	FERBITA BEREFERBITA BEREFERBITA

```
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()))
   result = find_equilibrium_position(N, A)
   print(result)
RESULT
 5 / 5 Test Cases Passed | 100 %
```