ો	Dogo Por Sorres
512A3BR21	TAILS SERVICES TO A SHELD CEST AND A SH
.S`	TAILS THE STUDENT REPORT
DET.	
BREDEI	AILS AST 1925 1 1987 1 1925 1
Na Na	me sci skri skri skri skri skri skri skri skr
~ _	WALLET ME AND A STATE OF THE ADDRESS
300	
,	ERIMENT A A SECOLAR AREA SECOLA
EXP	ERIMENTAL ASCS ASSET
Title	ERIMENTAL SCS ARASCS AR
EO	UILIBRIUM 355 355 355 355 355 355 355 355 355 35
A PROPERTY DE	ERIMENT 1 A 3 BR 13 C S 1 A 3 BR 13 C S 1 1 A 3 BR 13 C S 1 1 A 3
30 De	Scription 25 - SV - S
`	You are given an array A of N integers. An equilibrium position is a position where the sum of all integers on its left is equal to the sum of all integers on its left is equal to the sum of all integers on its right in the array A. Print the index of the equilibrium position.
223057	Note: For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT FOUND" without
جي (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.
- 0	The array is 1 indexed.
512438	
, I	Input Format:
~ ~	The input consists of two lines:
*3BR136	The first line contains an integer denoting N.
-	The first line contains an integer denoting N. The second line contains N space-separated integers denoting the elements of the array A.
-57	nput will be read from the STDIN by the candidate
273CS) (Output Format: Print the index of the equilibrium position. If no index is found, print "NOT FOUND"
·	Print the index of the equilibrium position. If no index is found, print "NOT FOUND"
3R :	Sample Input
512438	5
7	24733
3BR736	Sample Output
^ر ئ (
So.	urree Code: - S
00	urce Code: 3C5) 38R2 S71A3 S72C51A3 S73C51A3 S73
	38x2 ³ 34x2 ³ 251243 25512438x2 ³ 25512438x2 ³ 2551243 255
	38K13C5TV ARVACETVAST ARVACETV
	Service Control of the control of th

```
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 3BRITA SESTINA SESTI
                       A = list(map(int, input().split()))
                       result = find_equilibrium_position(N, A)
                       print(result)
RESULT
            5 / 5 Test Cases Passed | 100 %
```