	Dogo S S S S S S S S S S S S S S S S S S
2,476	ALADSAON AT ABAN AND STUDENT REPORT ABANAS SAOTA ABANASAN ABANASAN
05401.72	2281 SIDDENT REPORT
DE N	Stalls 128120540 11728120540 128120540 17281205400000000000000000000000000000000000
DE	TAILS TO THE TAILS TO THE TAILS TO THE TAIL THE
< The state of the	STUDENT REPORT  STAILS  1.281.405.401.7.2281.405.401.401.401.401.401.401.401.401.401.401
	MOHAMMED JUNED M
LADSAOTE	Roll Number 1 <sup>th</sup> 1th
LAN	22BI24DS401-T
EX	22BI24DS401-T  PERIMENT 12B124DS401 122B124DS401 122B124D
√ V Titl	le psaol psaol parabsaol parabsaol parabsaol parabsaol parabsaol parabsaol parabsaol psaol parabsaol psaol parabsaol
E	TOURN PRIMA AND AND AND AND AND AND AND AND AND AN
512AD 5AO	All Shans and the contract of the state of t
12ADS [	Description 28th 128 128th 28th 28th 28th 28th 28th 28th 28th
	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
,01.T.22P	of all integers on its right in the array A. Print the index of the equilibrium position.
xon.i.	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.
	The array is 1 indexed
apsi	
28124054	Input Format:  The input consists of two lines:
	The input candida of two lines.
5401.12	The first line contains an integer denoting N.
SAC	The first line contains an integer denoting N.  The second line contains N space-separated integers denoting the elements of the array A.
Č,	Input will be read from the STDIN by the candidate
2281240	Output Format:  Print the index of the equilibrium position. If no index is found, print "NOT FOUND"
\V	Print the index of the equilibrium position. If no index is found, print "NOT FOUND"
205A01-1	
×	24733
228120	Sample Output
222	3
S	Source Code: 22812 ADS AD 1. 122 BY AD 1.
	The state of the s
	228 DEAD SADIN SAD
	Sample Output  3  Source Code: Page Page Page Page Page Page Page Page
	BILLIE CHAIL CAN SERVICE CAN S
	22 SAN JASIN JANI JASIN JANI JASIN JANI JASIN JANI JASIN JANI JASIN JANI JASIN JANI JANI JANI JANI JANI JANI JANI JA

```
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()))
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    228 228 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 128 240 - 12
                       result = find_equilibrium_position(N, A)
                       print(result)
RESULT
           5 / 5 Test Cases Passed | 100 %
```