	Dogo √ √ √ √ √ √ √ √ √ √ √ √ √ √ √ √ √ √	
c,N,C	St. St. St. CHIDENT DEDON'T ST. St	
Biechic	STUDENT REPORT  STUDENT REPORT  STUDENT REPORT  LIMITED TO SELD STATE OF THE SELD ST	EMP
	afecti. The section of the section o	5
DE	ETAILS  Name Notice 1 Cost 1 Control of the Notice 1 Cost	
78'50 I	States Name Notice 1.52 1085 Temperate the Cartest of the Cartest	SE
	VENOVARDITAN SHETTI	30
echicst	Roll Number ( N	
ec,		MPB
EX	TEMPBTech-CSE085  KPERIMENT:  Bell Number of the Color of	
Tit	tle horizontalia de la companya de l	4
	EOUIDADINA CELOS. THUSA, CELOS.	SEO8.
\ \ \ \	The strain of th	
CSE 1	Description Republic Carter Andrew	ď
	TEMPBTech-CSE085  KPERIMENT  Lie  EQUILIBRIUM  Description  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	NPBTec)
EMPBIE	<b>Note</b> : 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.	£085 T
5	The diray is a indexed.	
SEOSS	Input Format:	Stechic
	The input consists of two lines:	200
Palech,		
387	The second line contains N space-separated integers denoting the elements of the array A.	STEM
	Input will be read from the STDIN by the candidate	50
085 [4]		
08	Print the index of the equilibrium position. If no index is found, print "NOT FOUND"	ichi cst
<		,,,
echics	5	\hat{\chi}
(°	24733	LEMP P
08	Sample Output	)
LEWI	3	~ (%)
		Castle .
;	Source Code: Paris Control Con	.>
	TEM, ELECTIC FORE WASHING CEPTOR.	1,20
	Source Code: Lind Hearth Carlos Lind Hard Collections of the Hard Land Hard Carlos Linds Lind Hard Carlos Linds Linds Land Hard Carlos Linds Linds Linds Linds Linds Land Hard Carlos Linds Linds Linds Land Land Land Land Land Land Land Land	While
	ARBITAL CELOS CENTES CONTRACTOR C	
	AK SELIES INTERIOR SELIES SELI	50 J
	Source Code: The state of the s	5

```
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 \%
```

the car

LECTION NO.

esta reco

SEO8

MR