	Logo 1.
NA.	ETAILS  Name 3 CID ON 3 BER 3 CID ON
CDOL	STUDENT REPORT  STUDENT REPORT
	CDOWN SESSEL CDOWN SESSEL CDOWN SESSEL CDOWN SESSEL CDOWN SESSEL WAS SELL COOM
BRE	ETAILS OF SOLO SERVICE SERVICE SOLO SERVICE SOLO SERVICE SOLO SERVICE SOLO SERVICE SOLO SERVICE SERVICE SOLO SERVICE SOLO SERVICE
XX	Name 3CD 3BRILL DOWN 3SCD 3BRILL DOWN 3SCD 3BRILL DOWN
	KEERIHI
3R23CD	3BR23CD044  XPERIMENT  A  B  B  B  B  B  B  B  B  B  B  B  B
p.	3BR23CD044
E	XPERIMENT SECOND
TAROOL	itle 13°CV 1
	EQUILIBRIUM 3CION ARCION ARCIO
N	XPERIMENT AND SUDDAM SERVISCIDAM SERVISCID
xA 3BK	SPERIMENT  A SHELL SCHOLAR SHE
	You are given an array A or N integers. An equilibrium position is a position where the sum of all integers on its left is equal to the sum
R23cDS	
3223	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.
COOAA	
	Input Format: The input consists of two lines:
c	The input consists of two lines:
xA 3BRI	The first line contains an integer denoting N.
× <sup>lx</sup>	The first line contains an integer denoting N.  The second line contains N space-separated integers denoting the elements of the array A.
00	Input will be read from the STDIN by the candidate
5R13CDC	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"
0	Sample Input
SOOAA	
	24733
3822	Sample Output
D*	3
	Source Code: Solver Service Solver Service Solver Service Solver Service Solver Service Service Solver Service Solver Service
	Source Code: 3CD AA 3CD
	Source Code: 3CDV 3HR13CDQAA3HR13CQAA3AA3CQAA3AA3CQAA3AAA3CQAA
	3° 3clor 34P23 - ONA 3° OSCOO AND CON AND CONTRACTOR OF CO
	Source Code: 3CD AA 3BR 3CD CAA 3BR 2CD CA
	Source Code: 3CD 3BR 2CD AA 3BR 2

```
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 %
```

8RJ3

Scoo.

BRI

AAA S

CDO

BRIS

NA 3

200