Logo	
Name, KURPASELEDON, KURPASELED	WIRDSHEET LEED
FIRE STORES THE STORES OF THE	-01 KD
118132 (001 123EEE 1 1418E 1 1413EE 10140	FIRS
DETAILS 32 EEE LUBY EEE CO LUBY 32 EEE ALUBY	EEOO
Name, KIND SELECTION KURD SELECTION	EFOOT KUB23EFFOOT KUB23EFFOOT KU
AISHWARYA	
Roll Number of Roll N	23t KUB23EE001 KUB23EE
KUB23EEE001	F1853,
EXPERIMENT SEED TO SEED SEED SEED SEED SEED SEED SEED SEE	25001
EXPERIMENT 3 LELLON KUR 3 LELLO	1873EE
EQUILIBRIUM 11873 COLLEGE AND	201 AD SEEELO
TABLES (1800) TO 325EEF 37 AUB! (1803) (1803) (1803) TO 325EEF	TAR.
EXPERIMENT 3 LL LIDO L	LOOT KUB? 3EE HOOT
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 ed	qual to the sum
of all integers on its right in the array A. Print the index of the equilibrium position.	,
Note: For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT For quotes.	OUND" without
The array is 1 indexed.	JBL
, kos	
Input Format:	EFFOOT
The first transfer of the Property	EE
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.	\K11873 ⁵
Input will be read from the STDIN by the candidate)
Output Format: Print the index of the equilibrium position of the index is found print "NOT FOUND"	,00
Print the index of the equilibrium position. If no index is found, print "NOT FOUND"	,23EE00
Sample Input	>
Sample input 5	
24733	- Baise
Sample Output	F
3	
	2. Jake Harry
Source Code: 35th, Fifth May 3th 100, Out May 35th Out Ma	OF THE BOY
Le Calebra Langi Calebra Caleb	William William
Source Code: 35 FFF On File of	De la
Source Code: 35EFF 1001 KIB13EFF1001 KIB13EF	A STATE OF THE PARTY OF THE PAR
Source Code: 3-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	SO STATE OF
Top Top The Control of the Control o	\$000 C \$0.50

```
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()))
                                                                                                   LU823EE001 KU823
   result = find_equilibrium_position(N, A)
   print(result)
RESULT
 0 / 5 Test Cases Passed | 0 %
```