	Logo
, cDobs 35	ETAILS  Name 3 CIU DENT REPORT  PAVAN KUMAR M JANGANI
CDOO	THE STORES STORES TO SEE THE SECOND S
	STUDENT REPORT  COLORS AREASCHOOLS AREASCH
D	ETAILS 18 18 18 18 18 18 18 18 18 18 18 18 18
650	Name 3 CDS 3 SEPT 3 COST 3 SEPT 3 SEPT 3 COST 3 SEPT 3 SEP
	PAVAN KUMAR M JANGANI
ac Dol	Roll Number 13 653 200 8813 253 200 2813
3R23CD0	3BR23CD065
F.	SOCIATION OF SPECIAL S
	3BR23CD065  (PERIMENT)  (PERIM
,cloosti	EQUIEIBRIUM  TO Description 15 3 April 200 16 April 200 1
	EQUEIBRIUM 300 340 500 340 500 340 500 340 500 340 500 340 500 340 500 340 500 340 500 340 500 340 500 340 500
SSARRIS	Description Astronomy
(5)	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 right in the array A. Print the index of the equilibrium position.
3R23CD01	Note: For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT FOUND" without
PK.	
2	The array is 1 indexed.
Choops	
Ç	Input Format:  The input consists of two lines:
\n^3	The input solicities of the lines.
53BR136	The first line contains an integer denoting N.  The second line contains N space-separated integers denoting the elements of the array A.
0	The second line contains N space-separated integers denoting the elements of the array A.
204	Input will be read from the STDIN by the candidate
5R13CDOK	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"
638	Sample Input
chots of	5
)	24733
3BR135	Sample Output
36,	3
	Source Code: 3CD 3APA CODE STATE SCOT SCOT STATE SCOT STATE SCOT STATE SCOT STATE SCOT SCOT STATE SCOT STATE SCOT SCOT STATE SCOT STATE SCOT STATE SCOT SCOT SCOT SCOT STATE SCOT SCOT SCOT SCOT SCOT SCOT SCOT SCOT
	Source Code: 3CD SARPACIO SARP
	Source Code: 300 Source
	3° COSTO COS
	AREA CONTRACTOR OF THE PROPERTY OF THE PROPERT

```
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 %
```