	Logo	
38	ELogo STUDENT REPORT PETAILS Name AKILA MANASA Roll Number ARA Roll Number	
3822°	STUDENT REPORT PETAILS Name AKII A MANASA	3AJO
	3823A, 3824, 1008, 334, 008, 345, 008, 344, 346, 346, 346, 346, 346, 346, 346	
,00	PÊTAILS 300 3823 000 300 300 300 300 300 300 300 300 3	0
23A,	Name 38th 1083 23th 38th 38th 1083 3th 3th 1083 18th	1008
	AKILA MANASA	
100838	Roll Numbers 38th 38th 38th 38th 38th 38th 38th	0
100		8381
F	3BR23AI008 EXPERIMENT, R2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	,
38PT	EXPERIMENT 18 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	O.P.
8	FOLH BRILIM	3553
	1083 3A108 3A23A1 3AB23 1083 3A108 3AB23A 083A1 08	
5R23A10	Description 22 100 3th 1822 1822 1822 1822 1822 1822 1822 182	.008
b `	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	3A1008
	adm of all integers of its right in the array A. Thirt the index of the equilibrium position.	
3A1008	Note : For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT FOUND" without quotes.	00838
	The array is 1 indexed.	~
108 3BR	$\mathfrak{f}_{\mathfrak{f}}$	2
200	Input Format:	3BR2?
	The input consists of two lines:	
38R23R	The first line contains an integer denoting N.	10
D*	The second line contains N space-separated integers denoting the elements of the array A.	223A10
0	Input will be read from the STDIN by the candidate	
23A100	Output Format:	A1008
	Print the index of the equilibrium position. If no index is found, print "NOT FOUND"	Mos
38	Sample Input	
1008 38	5	388
	24733	3800
22	Sample Output	,
36	3	3832
	_3,	33~
	Source Code: 344231 34400 34420 34400	28
	33 LOV 38	3246
	Source Code: 3842, 341008 342,	
	Source Code: 38 22 3 100 8 30 100 8 3	28
	2823AL 38E22 38E22 38E22 38E223 38E223 38E223AL 38E2EEEE	1888°
		,

```
{\tt 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
  0 / 5 Test Cases Passed | 0 %
```