| | □Logo | |
|------------------|--|---------|
| N. | SEN SERVICE DEDON'T SENSON SENSON SERVICE SENSON SENSON SERVICE SENSON SENSON SERVICE SENSON SENSON SERVICE SENSON SERVICE SENSON SERVICE SENSON SERVICE SENSON S | |
| 23B/C | STUDENT REPORT | BRI |
| | 382° 16° 36° 38° 16° 16° 16° 16° 16° 16° 16° 16° 16° 16 | V |
| D | DETAILS 23th 3th 160 3 25th 3th 3th 3th 3th 3th 3th 3th 3th 3th 3 | |
| SKJ3, | STUDENT REPORT Name, 67 SYEDA AALIYA FATIMA | 3811 |
| | OTED/TO CETT/TO THIS | 5 |
| | | C |
| BAILOS | 200224142 | 1162 |
| | 3BR23AI162 EXPERIMENT 3BR23AI162 | ~ |
| 3 ⁶ 7 | EXPERIMENTS TO SEE SHEET SO SEE SHEET STATE STAT | o c |
| 3BT | itle EQUILIBRIUM TO 34E134 SERISE SERIES | J. 38/ |
| | EQUILIBRIUM | |
| 3BR23A | itle EQUILIBRIUM Description R102 34423 A1102 34423 | À |
| 36, | 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 | 3R23A |
| | of all integers on its right in the array A. Print the index of the equilibrium position | |
| 23A1162 | Note: For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT FOUND" without | .67 |
| 200 | γ | 3A1767 |
| 25 | | |
| 1162 3BE | | 62 3B |
| 2. | | 6 |
| 3 | The input consists of two lines. | |
| 13BR231 | The first line contains an integer denoting N. | 223 |
| V | \$\frac{1}{2} | 3BR231 |
| 1,00 | nput will be read from the STDIN by the candidate | |
| ,223A10 | Output Format: | 23A176 |
| | Print the index of the equilibrium position. If no index is round, print (NOT FOOND) | L |
| ,A16238 | Sample Input | |
| PIJO | 5 | 18 K3 |
| | 24733 | EL |
| 3822 | Sample Output | 9 |
| .b. | 3 | 388 |
| | Source Code: 3 P. 1 3 P. 2 3 P | 5 |
| | Source Code: 3 Pil | 13 |
| | 38 FLY NOV 3 TO SELLY SE | Bar |
| | 36, 24/24, 138/36, 173/4, 170/2, 38/36, 178/36, 178/36, 178/36, 178/36, 178/36, 178/36, 178/36, 178/36, 178/36 |) * |
| | Source Code: 34, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12 | (2) |
| | ARISE LIGHT MARKET LIGHT AND AREA TO A | Self 3r |
| | 30 10 10 10 10 10 10 10 10 10 10 10 10 10 | |

```
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 %
```