| DETAILS Name UP 2 SERVE LUP 2 |
|--|
| STUDENT REPORT UR23CSEDAS LUB23CSEDAS LUB |
| STODENT REPORT |
| DETAILS SEON Name UP 27 SELDAT KUP 23 SELDA |
| DETAILS SEON TOWNS TO THE SECOND SECO |
| DETAILS SEOMS WAR STORY TO SEE OF SECONS AND SECONS AN |
| JASON KENNETH N |
| KUB23CSE045 |
| KUB23CSE045 |
| EXPERIMENTS CELONS FOR THE STATE OF THE STAT |
| 25 - 11 - 15 - 15 - 15 - 15 - 15 - 15 - |
| EQUÍLIBRIUM LUBY SELON LUBY SELON SE |
| 1823 Chr. 40 32560 F1823 Chr. 40813 1823Cr. |
| Description For Annual Control of the Control of th |
| 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. |
| 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 FOUND" without quotes. The array is 1 indexed. |
| The array is 1 indexed. |
| |
| Input Format: |
| |
| 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 first line contains an integer denoting N. The second line contains N space-separated integers denoting the elements of the array A. |
| ξ |
| Input will be read from the STDIN by the candidate Output Format: Drive the index of the applibly interpretable index is found again. MOT FOUND! |
| Output Format: Print the index of the equilibrium position. If no index is found, print "NOT FOUND" |
| |
| Sample Input 5 |
| 5 2 4 7 3 3 3 <u>A E</u> |
| ൂ ^{ട്ട} Sample Output |
| Sample Output 3 |
| |
| Source Code: 325 E. LUBL 25 ELDAS LUBL 25 EL |
| The state of the s |
| The state of the s |
| Source Code: 32-55 L. LUB23-5-510 A.5 LUB23-5- |
| Source Code: 3.55 To 1.10 Part of the first |
| The stops of the stop of the stops of the stops of the stops of the stop of th |

```
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 %
```