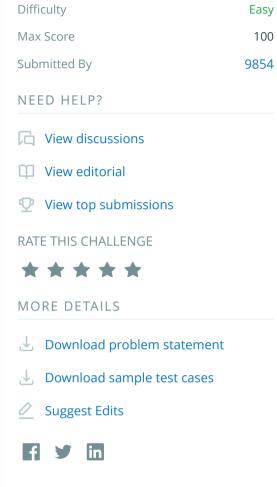
Author

[deleted]



## **Left Rotation** ★

·	ation on an array of size $m{n}$ shifts each of the array's elements $m{1}$ unit to the left. Given an integer, $m{d}$ ,
-	at many steps left and return the result.
Example $d=2$	
$egin{aligned} a = z \ arr = [1,2,3,4,5] \end{aligned}$	5]
-	$m{rr'} = [3,4,5,1,2]$ .
Function Descript	
_	eLeft function in the editor below.
•	following parameters:
• int d: the amour	
• int arr[n]: the ar	ray to rotate
Returns	
• int[n]: the rotate	ed array
Input Format	
The first line contai	ns two space-separated integers that denote $m{n}$ , the number of integers, and $m{d}$ , the number of left
rotations to perfor	n.
The second line co	ntains $m{n}$ space-separated integers that describe $m{arr}[]$ .
Constraints	
• $1 \leq n \leq 10^5$	
• $1 \leq d \leq n$	
• $1 \leq a[i] \leq 10^6$	
Sample Input	
5 4 1 2 3 4 5	
Sample Output	
5 1 2 3 4	
J I Z J 4	
Explanation	
To perform $oldsymbol{d}=4$	eft rotations, the array undergoes the following sequence of changes:





```
* Complete the 'rotateLeft' function below.
      \star The function is expected to return an INTEGER_ARRAY.
      * The function accepts following parameters:
      * 1. INTEGER d
14
      * 2. INTEGER_ARRAY arr
      */
17
     vector<int> rotateLeft(int d, vector<int> arr) {
     }
     int main()
     {
         ofstream fout(getenv("OUTPUT_PATH"));
24
         string first_multiple_input_temp;
         getline(cin, first_multiple_input_temp);
27
         vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));
                                                                     Line: 103 Col: 1
                                                                          Submit Code
                                                            Run Code

↑ Upload Code as File

                    Test against custom input
```