**Elite-S020-Arrays**

Solved Challenges **0/2**

Back To Challenges List

## Array Rotation Left R times

**ID:11096** **Solved By 718 Users**

You must implement the function **rotate(int arr[],int N,int R)** which accepts an integer array **arr** with it's size **N** and an integer **R** as the input. The function must rotate the array by shifting it R times to the left.

**Boundary Condition(s):**
1 <= N <= 10^5
1 <= Array element value <= 10^4
1 <= R <= 10^8

**Example Input/Output 1:**
Input:
4
10 20 30 40
1000002

Output:
30 40 10 20

Explanation:
Here R = **2**
After the **first** left-rotation, the integers in the array become 20 30 40 10
After the **second** left-rotation, the integers in the array become 30 40 10 20
Hence the output is 30 40 10 20

**Example Input/Output 2:**
Input:
7
76 74 18 17 45 29 11
5

Output:
29 11 76 74 18 17 45

**Max Execution Time Limit: 100 millisecs**

Ambiance

C ( gcc 8.x)

Reset

```
#include<stdio.h>
```

```
void rotate(int arr[],int N,int R)
{
    R = R%N;

    reverse(arr,0,N-1);
    reverse(arr,0,N-R-1);
    reverse(arr,N-R,N-1);
}

int reverse(int arr[],int start,int end)
{
    while(start<end)
    {
        int temp = arr[end];
        arr[end] = arr[start];
        arr[start] = temp;
        start++;
        end--;
    }
}
```

```
int main()
{
    int N,R;
    scanf("%d",&N);
    int arr[N];
    for(int index=0; index<N; index++)
    {
        scanf("%d",&arr[index]);
    }
    scanf("%d",&R);
    rotate(arr,N,R);
    for(int index=0; index<N; index++)
    {
        printf("%d ",arr[index]);
    }
}
```

**Code did not pass the execution**

**Hello.c: In function ‘reverse’:**
**Hello.c:14:9: error: expected expression before ‘int’**

```
        int temp = arr[end];
        ^~~
Hello.c:16:22: error: ‘temp’ undeclared (first use in this function)
        arr[start] = temp;
                     ^~~~
Hello.c:16:22: note: each undeclared identifier is reported only once for each function it app
ears in
Hello.c: At top level:
Hello.c:20:1: error: expected identifier or ‘(’ before ‘}’ token
 }
 ^
```

Save Run

Run with a custom test case (Input/Output)