

Elite-S020-ArraysSolved Challenges **1/2**[Back To Challenges List](#)**Array Rotation Right R times****ID:11097 Solved By 731 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 right.

Boundary Condition(s): $1 \leq N \leq 10^5$ $1 \leq \text{Array element value} \leq 10^4$ $1 \leq R \leq 10^8$ **Example Input/Output 1:**

Input:

10

10 20 30 40 50 60 70 80 90 100

3

Output:

80 90 100 10 20 30 40 50 60 70

Explanation:

Here $R = 3$ After the **first** right-rotation, the integers in the array become 100 10 20 30 40 50 60 70 80 90After the **second** right-rotation, the integers in the array become 90 100 10 20 30 40 50 60 70 80After the **third** right-rotation, the integers in the array become 80 90 100 10 20 30 40 50 60 70

Hence the output is 80 90 100 10 20 30 40 50 60 70

Example Input/Output 2:

Input:

5

45 78 12 98 56


10004

Output:

78 12 98 56 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,R-1);
    reverse(arr,R,N-1);
}
void reverse(int arr[],int start,int end)
{
    while(start<end)
    {
        int temp=arr[start];
        arr[start]=arr[end];
        arr[end] =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]);
    }
}
```

Save

Run

☐ Run with a custom test case (Input/Output)