***Left Rotate an Array by D places***

Given an array of integers **arr[]** of size **N** and an integer, the task is to rotate the array elements to the **left** by **d** positions.

**Examples:**

***Input:***  
*arr[] = {1, 2, 3, 4, 5, 6, 7}, d = 2*  
***Output:****3 4 5 6 7 1 2*

***Input:****arr[] = {3, 4, 5, 6, 7, 1, 2}, d=2*  
***Output:****5 6 7 1 2 3 4*

***Naive:***

C++Java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class GFG

{

static void lRotateOne(int arr[], int n)

{

int temp = arr[0];

for(int i = 1; i < n; i++)

{

arr[i - 1] = arr[i];

}

arr[n - 1] = temp;

}

static void leftRotate(int arr[], int d, int n)

{

for(int i = 0; i < d; i++)

{

lRotateOne(arr, n);

}

}

public static void main(String args[])

{

int arr[] = {1, 2, 3, 4, 5}, n = 5, d = 2;

System.out.println("Before Rotation");

for(int i = 0; i < n; i++)

{

System.out.print(arr[i]+" ");

}

System.out.println();

leftRotate(arr, d, n);

System.out.println("After Rotation");

for(int i = 0; i < n; i++)

{

System.out.print(arr[i]+" ");

}

}

}

**Output:**

Before Rotation

1 2 3 4 5

After Rotation

3 4 5 1 2

**ReversaL Method:**

C++Java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class GFG

{

static void leftRotate(int arr[], int d, int n)

{

reverse(arr, 0, d - 1);

reverse(arr, d, n - 1);

reverse(arr, 0, n - 1);

}

static void reverse(int arr[], int low, int high)

{

while(low < high)

{

int temp = arr[low];

arr[low] = arr[high];

arr[high] = temp;

low++;

high--;

}

}

public static void main(String args[])

{

int arr[] = {1, 2, 3, 4, 5}, n = 5, d = 2;

System.out.println("Before Rotation");

for(int i = 0; i < n; i++)

{

System.out.print(arr[i]+" ");

}

System.out.println();

leftRotate(arr, d, n);

System.out.println("After Rotation");

for(int i = 0; i < n; i++)

{

System.out.print(arr[i]+" ");

}

}

}

**Output:**

Before Rotation

1 2 3 4 5

After Rotation

3 4 5 1 2

**Efficient Approach:**

C++Java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class GFG

{

static void leftRotate(int arr[], int d, int n)

{

int temp[] = new int[d];

for(int i = 0; i < d; i++)

{

temp[i] = arr[i];

}

for(int i = d; i < n; i++)

{

arr[i - d] = arr[i];

}

for(int i = 0; i < d; i++)

{

arr[n - d + i] = temp[i];

}

}

public static void main(String args[])

{

int arr[] = {1, 2, 3, 4, 5}, n = 5, d = 2;

System.out.println("Before Rotation");

for(int i = 0; i < n; i++)

{

System.out.print(arr[i]+" ");

}

System.out.println();

leftRotate(arr, d, n);

System.out.println("After Rotation");

for(int i = 0; i < n; i++)

{

System.out.print(arr[i]+" ");

}

}

}

**Output:**

Before Rotation

1 2 3 4 5

After Rotation

3 4 5 1 2