

1. Write a function `rotate(ar[], d, n)` that rotates `arr[]` of size `n` by `d` elements.
2. Rotate the array A clockwise by `d` units.
3. Rotate the array A anticlockwise by `d` units.
4. Block swap algorithm for array rotation
Write a function `rotate(ar[], d, n)` that rotates `arr[]` of size `n` by `d` elements.

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Rotation of the above array by 2 will make array

3	4	5	6	7	1	2
---	---	---	---	---	---	---

Algorithm :

Initialize $A = \text{arr}[0..d-1]$ and $B = \text{arr}[d..n-1]$

1) Do following until size of A is equal to size of B

- a) If A is shorter, divide B into B_1 and B_r such that B_r is of same length as A. Swap A and B_r to change AB_1B_r into B_rB_1A . Now A is at its final place, so recur on pieces of B.
- b) If A is longer, divide A into A_1 and A_r such that A_1 is of same length as B. Swap A_1 and B to change A_1A_rB into BA_rA_1 . Now B is at its final place, so recur on pieces of A.

2) Finally when A and B are of equal size, block swap them.

5. Given an array, cyclically rotate the array clockwise by one.

Examples:

Input: `arr[] = {1, 2, 3, 4, 5}`

Output: `arr[] = {5, 1, 2, 3, 4}`

6. Search an element in a sorted array. Devise a way to find an element in the rotated array in $O(\log n)$ time.
7. Given an array that is sorted and then rotated around an unknown point. Find if array has a pair with given sum 'x'. It may be assumed that all elements in array are distinct.

Examples:

Input: $\text{arr}[] = \{11, 15, 6, 8, 9, 10\}$, $x = 16$

Output: true

There is a pair (6, 10) with sum 16

Input: $\text{arr}[] = \{11, 15, 26, 38, 9, 10\}$, $x = 35$

Output: true

There is a pair (26, 9) with sum 35

Input: $\text{arr}[] = \{11, 15, 26, 38, 9, 10\}$, $x = 45$

Output: false

There is no pair with sum 45.

8. Find maximum value of $\text{Sum}(i * \text{arr}[i])$ with only rotations on given array allowed. Given an array, only rotation operation is allowed on array. We can rotate the array as many times as we want. Return the maximum possible of summation of $i * \text{arr}[i]$.

Examples:

Input: $\text{arr}[] = \{1, 20, 2, 10\}$

Output: 72

We can 72 by rotating array twice.

$\{2, 10, 1, 20\}$

$20*3 + 1*2 + 10*1 + 2*0 = 72$

Input: $\text{arr}[] = \{10, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$;

Output: 330

We can 330 by rotating array 9 times.

$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$;

$0*1 + 1*2 + 2*3 \dots 9*10 = 330$

9. Maximum sum of $i * \text{arr}[i]$ among all rotations of a given array. Given an array $\text{arr}[]$ of n integers, find the maximum that maximizes sum of value of $i * \text{arr}[i]$ where i varies from 0 to $n-1$.

Examples:

Input : arr[] = {8, 3, 1, 2}

Output : 29

Explanation : Let us see all rotations

$\{8, 3, 1, 2\} = 8*0 + 3*1 + 1*2 + 2*3 = 11$

$\{3, 1, 2, 8\} = 3*0 + 1*1 + 2*2 + 8*3 = 29$

$\{1, 2, 8, 3\} = 1*0 + 2*1 + 8*2 + 3*3 = 27$

$\{2, 8, 3, 1\} = 2*0 + 8*1 + 3*2 + 1*3 = 17$

Input : arr[] = {3, 2, 1}

Output : 8

10. Find the Rotation Count in Rotated Sorted array.

Consider an array of distinct numbers sorted in increasing order. The array has been rotated (clockwise) k number of times. Given such an array, find the value of k.

Examples:

Input : arr[] = {15, 18, 2, 3, 6, 12}

Output: 2

Explanation : Initial array must be {2, 3, 6, 12, 15, 18}. We get the given array after rotating the initial array twice.

Input : arr[] = {7, 9, 11, 12, 5}

Output: 4

Input: arr[] = {7, 9, 11, 12, 15};

Output: 0

11. Quickly find multiple left rotations of an array.

Given an array of size n and multiple values around which we need to left rotate the array. How to quickly find multiple left rotations?

Examples:

Input : arr[] = {1, 3, 5, 7, 9}

k1 = 1

k2 = 3

k3 = 4

k4 = 6

Output : 3 5 7 9 1

```
7 9 1 3 5
9 1 3 5 7
3 5 7 9 1
```

Input : arr[] = {1, 3, 5, 7, 9}
k1 = 14

Output : 9 1 3 5 7

12. Find the minimum element in a sorted and rotated array.

A sorted array is rotated at some unknown point, find the minimum element in it.
Following solution assumes that all elements are distinct.

Examples:

Input: {5, 6, 1, 2, 3, 4}
Output: 1

Input: {1, 2, 3, 4}
Output: 1

Input: {2, 1}
Output: 1

13. Rotate the matrix right by K times. Given a matrix of size N*M, and a number K.

We have to rotate the matrix K times to the right side.

Examples:

Input : N = 3, M = 3, K = 2
12 23 34
45 56 67
78 89 91

Output : 23 34 12
56 67 45
89 91 78

Input : N = 2, M = 2, K = 2
1 2

3 4

Output : 1 2

3 4

14. Reversal algorithm for right rotation of an array.

Given an array, right rotate it by k elements.

15. Find a rotation with maximum hamming distance.

Given an array of n elements, create a new array which is a rotation of given array and hamming distance between both the arrays is maximum.

[Hamming distance](#) between two arrays or strings of equal length is the number of positions at which the corresponding character(elements) are different.

Note: There can be more than one output for the given input.

Examples:

Input : 1 4 1

Output : 2

Explanation:

Maximum hamming distance = 2.

We get this hamming distance with 4 1 1

or 1 1 4

input : N = 4

2 4 8 0

output : 4

Explanation:

Maximum hamming distance = 4

We get this hamming distance with 4 8 0 2.

All the places can be occupied by another digit.

Other solutions can be 8 0 2 4, 4 0 2 8 etc.

16. Queries on Left and Right Circular shift on array

Given an array A of N integers. There are three type of type of commands:

1 x : Right Circular Shift the array x times. If an array is $a[0], a[1], \dots, a[n-1]$, then after one right circular shift the array will become $a[n-1], a[0], a[1], \dots, a[n-2]$.

2 y : Left Circular Shift the array y times. If an array is $a[0], a[1], \dots, a[n-1]$, then after one right circular shift the array will become $a[1], \dots, a[n-2], a[n-1], a[0]$.

3 l r : Print the sum of all integers in the subarray a[l...r] (l and r inclusive).

Given Q queries, the task is execute each query.

Examples:

Input : n = 5, arr[] = { 1, 2, 3, 4, 5 }

query 1 = { 1, 3 }

query 2 = { 3, 0, 2 }

query 3 = { 2, 1 }

query 4 = { 3, 1, 4 }

Output : 12

11

Initial array arr[] = { 1, 2, 3, 4, 5 }

After query 1, arr[] = { 3, 4, 5, 1, 2 }.

After query 2, sum from index 0 to index
2 is 12, so output 12.

After query 3, arr[] = { 4, 5, 1, 2, 3 }.

After query 4, sum from index 1 to index
4 is 11, so output 11.

17. Count rotations divisible by 8.

Given a large positive number as string, count all rotations of the given number which are divisible by 8.

Examples:

Input: 8

Output: 1

Input: 40

Output: 1

Rotation: 40 is divisible by 8

04 is not divisible by 8

Input : 13502

Output : 0

No rotation is divisible by 8

Input : 43262488612

Output : 4

18. Sort a Rotated Sorted Array

You are given a [rotated sorted array](#) and your aim is to restore its original sort in place. Expected to use $O(1)$ extra space and $O(n)$ time complexity.

Examples:

Input : [3, 4, 1, 2]
Output : [1, 2, 3, 4]

Input : [2, 3, 4, 1]
Output : [1, 2, 3, 4]

19. Find element at given index after a number of rotations

An array consisting of N integers is given. There are several [Right Circular Rotations](#) of range $[L..R]$ that we perform. After performing these rotations, we need to find element at a given index.

Examples:

Input : arr[] : {1, 2, 3, 4, 5}
 ranges[] = { {0, 2}, {0, 3} }
 index : 1

Output : 3

Explanation : After first given rotation {0, 2}

 arr[] = {3, 1, 2, 4, 5}

 After second rotation {0, 3}

 arr[] = {4, 3, 1, 2, 5}

After all rotations we have element 3 at given index 1.

20. Split the array and add the first part to the end

There is a given an array and split it from a specified position, and move the first part of array add to the end.

12	10	5	6	52	36
5	6	52	36	12	10

Examples:

Input : arr[] = {12, 10, 5, 6, 52, 36}

k = 2

Output : arr[] = {5, 6, 52, 36, 12, 10}

Explanation : Split from index 2 and first part {12, 10} add to the end .

Input : arr[] = {3, 1, 2}

k = 1

Output : arr[] = {1, 2, 3}

Explanation : Split from index 1 and first part add to the end.

21. Print left rotation of array in $O(n)$ time and $O(1)$ space

Given an array of size n and multiple values around which we need to left rotate the array. How to quickly print multiple left rotations?

Examples:

Input : arr[] = {1, 3, 5, 7, 9}

k1 = 1

k2 = 3

k3 = 4

k4 = 6

Output : 3 5 7 9 1

7 9 1 3 5

9 1 3 5 7

3 5 7 9 1

Input : arr[] = {1, 3, 5, 7, 9}

k1 = 14

Output : 9 1 3 5 7

22. Rotate a Matrix by 180 degree.

Given a square matrix the task is that we turn it by 180 degrees in anti-clockwise direction without using any extra space.

Examples:

Input : 1 2 3

4 5 6

7 8 9

Output : 9 8 7

6 5 4

3 2 1

Input : 1 2 3 4

5 6 7 8

9 0 1 2

3 4 5 6

Output : 6 5 4 3

2 1 0 9

8 7 6 5

4 3 2 1

23. Minimum move to end operations to make all strings equal

Given n strings that are permutations of each other. We need to make all strings same with an operation that takes front character of any string and move it to end.

Examples:

Input : n = 2

arr[] = {"molzv", "lzmvo"}

Output : 2

Explanation: In first string, we remove first element("m") from first string and append it end. Then we move second character of first string and move it to end. So after 2 operations, both strings become same.

Input : n = 3

arr[] = {"kc", "kc", "kc"}

Output : 0

Explanation: already all strings are equal.

24. Minimum rotations required to get the same string

Given a string, we need to find the minimum number of rotations required to get the same string.

Examples:

Input : s = "geeks"

Output : 5

Input : s = "aaaa"

Output : 1

25. Lexicographically smallest rotated sequence

Write code to find lexicographic minimum in a circular array, e.g. for the array BCABDADAB, the lexicographic minimum is ABBCABDAD

Input Constraint: $1 < n < 1000$ Examples:

Input: GEEKSQUIZ

Output: EEKSQUIZG

Input: GFG

Output: FGG

Input : CAPABCQ

Output : ABCQCAP

26. Left Rotation and Right Rotation of a String

Given a string of size n , write functions to perform following operations on string.

1. Left (Or anticlockwise) rotate the given string by d elements (where $d \leq n$)

2. Right (Or clockwise) rotate the given string by d elements (where $d \leq n$).

Examples:

Input : s = "GeeksforGeeks"

d = 2

Output : Left Rotation : "eksforGeeksGe"

Right Rotation : "ksGeeksforGee"

Input : s = "qwertyu"

d = 2

Output : Left rotation : "ertyuqw"

Right rotation : "yuqwert"

27. Count rotations divisible by 4

Given a large positive number as string, count all rotations of the given number which are divisible by 4.

Examples:

Input: 8

Output: 1

Input: 20

Output: 1

Rotation: 20 is divisible by 4

02 is not divisible by 4

Input : 13502

Output : 0

No rotation is divisible by 4

Input : 43292816

Output : 5

5 rotations are : 43292816, 16432928, 81643292

92816432, 32928164

28. Check if a string can be obtained by rotating another string 2 places

Given two strings, the task is to find if a string can be obtained by rotating another string two places.

Examples:

Input : string1 = "amazon"
string2 = "azonam" // rotated anti-clockwise
Output : Yes

Input : string1 = "amazon"
string2 = "onamaz" // rotated clockwise
Output : Yes

1- There can be only two cases:

- a) Clockwise rotated
- b) Anti-clockwise rotated

2- If clockwise rotated that means elements are shifted in right.
So, check if a substring[2.... len-1] of string2 when concatenated with substring[0,1] of string2 is equal to string1. Then, return true.

3- Else, check if it is rotated anti-clockwise that means elements are shifted to left.
So, check if concatenation of substring[len-2, len-1] with substring[0....len-3] makes it equals to string1. Then return true.

4- Else, return false.

29. Rotate a matrix by 90 degree without using any extra space

Given a square matrix, turn it by 90 degrees in anti-clockwise direction without using any extra space.

Examples:

Input

1 2 3
4 5 6
7 8 9

Output:

3 6 9
2 5 8
1 4 7

Input:

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Output:
4 8 12 16
3 7 11 15
2 6 10 14
1 5 9 13

30. Find the Rotation Count in Rotated Sorted array

Consider an array of distinct numbers sorted in increasing order. The array has been rotated (clockwise) k number of times. Given such an array, find the value of k.

Examples:

Input : arr[] = {15, 18, 2, 3, 6, 12}

Output: 2

Explanation : Initial array must be {2, 3, 6, 12, 15, 18}. We get the given array after rotating the initial array twice.

Input : arr[] = {7, 9, 11, 12, 5}

Output: 4

Input: arr[] = {7, 9, 11, 12, 15};

Output: 0

31. A Program to check if strings are rotations of each other or not

Given a string s1 and a string s2, write a snippet to say whether s2 is a rotation of s1 using only one call to strstr routine? (eg given s1 = ABCD and s2 = CDAB, return true, given s1 = ABCD, and s2 = ACBD , return false)

Algorithm: areRotations(str1, str2)

1. Create a temp string and store concatenation of str1 to str1 in temp.
temp = str1.str1
2. If str2 is a substring of temp then str1 and str2 are rotations of each other.

Example:

str1 = "ABACD"

str2 = "CDABA"

temp = str1.str1 = "ABACDABACD"

Since str2 is a substring of temp, str1 and str2 are rotations of each other.

32. Check if all rows of a matrix are circular rotations of each other

Given a matrix of $n \times n$ size, the task is to find whether all rows are circular rotations of each other or not.

Input: mat[][] = 1, 2, 3

3, 1, 2

2, 3, 1

Output: Yes

All rows are rotated permutation of each other.

Input: mat[3][3] = 1, 2, 3

3, 2, 1

1, 3, 2

Output: No

Explanation : As 3, 2, 1 is not a rotated or circular permutation of 1, 2, 3

33. Rotate each ring of matrix anticlockwise by K elements

Given a matrix of order $M \times N$ and a value K, the task is to rotate each ring of the matrix anticlockwise by K elements. If in any ring elements are less than and equal K then don't rotate it.

Examples:

Input : k = 3

mat[4][4] = {{1, 2, 3, 4},
 {5, 6, 7, 8},
 {9, 10, 11, 12},
 {13, 14, 15, 16}}

Output: 4 8 12 16

3 10 6 15

2 11 7 14

1 5 9 13

Input : k = 2

```
mat[3][4] = {{1, 2, 3, 4},  
             {10, 11, 12, 5},  
             {9, 8, 7, 6}}
```

Output: 3 4 5 6

2 11 12 7

1 10 9 8

34. Inplace rotate square matrix by 90 degrees

Given an square matrix, turn it by 90 degrees in anti-clockwise direction without using any extra space.

Examples:

Input

1 2 3

4 5 6

7 8 9

Output:

3 6 9

2 5 8

1 4 7

Input:

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

Output:

4 8 12 16

3 7 11 15

2 6 10 14

1 5 9 13

35. Rotate Matrix Elements

Given a matrix, clockwise rotate elements in it.

Examples:

Input

```

1  2  3
4  5  6
7  8  9

```

Output:

```

4  1  2
7  5  3
8  9  6

```

For 4*4 matrix

Input:

```

1  2  3  4
5  6  7  8
9  10 11 12
13 14 15 16

```

Output:

```

5  1  2  3
9  10 6  4
13 11 7  8
14 15 16 12

```

36. Rotate bits of a number

Bit Rotation: A rotation (or circular shift) is an operation similar to shift except that the bits that fall off at one end are put back to the other end.

In left rotation, the bits that fall off at left end are put back at right end.

In right rotation, the bits that fall off at right end are put back at left end.

Example:

Let n is stored using 8 bits. Left rotation of n = 11100101 by 3 makes n = 00101111 (Left shifted by 3 and first 3 bits are put back in last). If n is stored using 16 bits or 32 bits then left rotation of n (000...11100101) becomes 00..0011100101000.

Right rotation of n = 11100101 by 3 makes n = 10111100 (Right shifted by 3 and last 3 bits are put back in first) if n is stored using 8 bits. If n is stored using 16 bits or 32 bits then right rotation of n (000...11100101) by 3 becomes 101000..0011100.

37. Turn an image by 90 degree

Given an image, how will you turn it by 90 degrees? A vague question. Minimize the browser and try your solution before going further.

An image can be treated as 2D matrix which can be stored in a buffer. We are provided with matrix dimensions and it's base address. How can we turn it?

For example see the below picture,

```
* * * ^ * * *  
* * * | * * *  
* * * | * * *  
* * * | * * *
```

After rotating right, it appears (observe arrow direction)

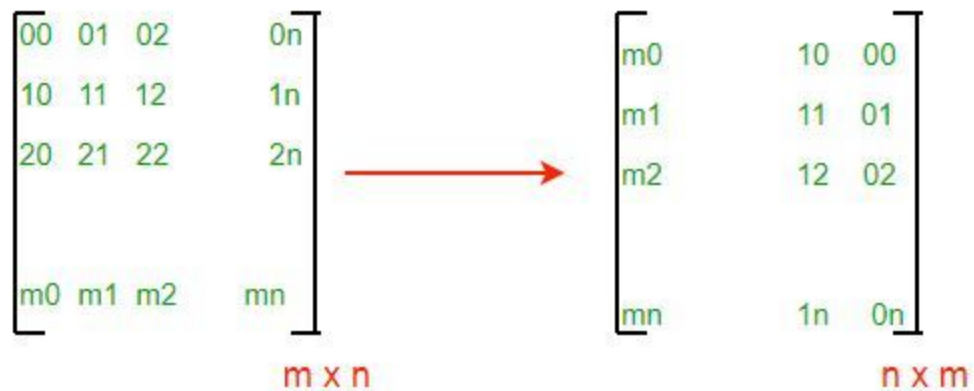
```
* * * *  
* * * *  
* * * *  
- - - >  
* * * *  
* * * *  
* * * *
```

The idea is simple. Transform each row of source matrix into required column of final image. We will use an auxiliary buffer to transform the image.

From the above picture, we can observe that

first row of source -----> last column of destination
second row of source -----> last but-one column of destination
so ... on
last row of source -----> first column of destination

In pictorial form, we can represent the above transformations of an (m x n) matrix into (n x m) matrix,



38. Lexicographically minimum string rotation

Write code to find lexicographic minimum in a circular array, e.g. for the array BCABDADAB, the lexicographic minimum is ABBCABDAD.

More Examples:

Input: GEEKSQUIZ
Output: EEKSQUIZG

Input: GFG
Output: FGG

Input: GEEKSFORGEEEKS
Output: EEKSFORGEEEKSG

Following is a simple solution. Let the given string be 'str'

- 1) Concatenate 'str' with itself and store in a temporary string say 'concat'.
- 2) Create an array of strings to store all rotations of 'str'. Let the array be 'arr'.
- 3) Find all rotations of 'str' by taking substrings of 'concat' at index 0, 1, 2..n-1. Store these rotations in arr[]
- 4) Sort arr[] and return arr[0].