Question: Given N, find the no. of factors

Any number which divides N is a factor

32: 1, 2, 4, 8, 16, 32

Quiz: Is 4 a factor of 24

4) 24 (

quotient = 6

remainder = 0 ✓

Yes! 24 is a factor

24 % 4 = 0 ⇒ Factor

↓

Modulo operator

Quiz: Check if i is a factor of N?

N % i == 0

↓

Remainder when N is divided by i

Quiz: No. of factors of N = 10

10: 1, 2, 5, 10 ⇒ 4 factors

Quiz: No. of factors of N = 12

12: 1, 2, 3, 4, 6 12 => 6 factors

Approach1:

i => N % i == 0

12 => [1, 12]

```
int count = 0;
for(i = 1; i <= N; i++){
        if( N % i == 0){
            count++;
        }
}
return count;
```

} => N iterations

1sec => 10^8 iterations/operations

$N = 10^8$ => 10^8 iterations => 1sec

Quiz: Time taken to run 10^9 iterations

$10^8 \Rightarrow$ 1 sec

$10^9 \Rightarrow$

1 sec $\Rightarrow$ 10^9 iters

2 sec $\Rightarrow$ 2×10^8 ⇐ [10^{16}]

3 sec $\Rightarrow$ 3×10^8 ⇐

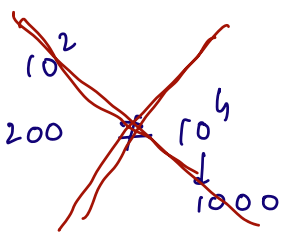
4 sec $\Rightarrow$ 4×10^8

⋮

10 sec $\Rightarrow$ 10×10^8 $\Rightarrow$ 10^9

$10^9 = 10 \times 10^8 \Rightarrow$ [10 seconds]

1 sec



Quiz: Time taken to run 10^{18} iterations

$10^8 \Rightarrow$ 1 sec

$10^{18} = x \times 10^8$

$x = \frac{10^{18}}{10^8} = 10^{18-8} = 10^{10}$

$10^{18} = 10^{10} \times 10^8$

1 sec

$\frac{a^b}{a^c} = a^{b-c}$

$10^{18} = 10^{10} \times 10^{8}$ (1 sec)

$= 10^{10} \times 1\text{sec} = 10^{10}$ seconds

$= 10000000000$ seconds

$\Downarrow$

317 years

Time to find # factors of 10^{18} $\Rightarrow$ 317 years

Approach 2:

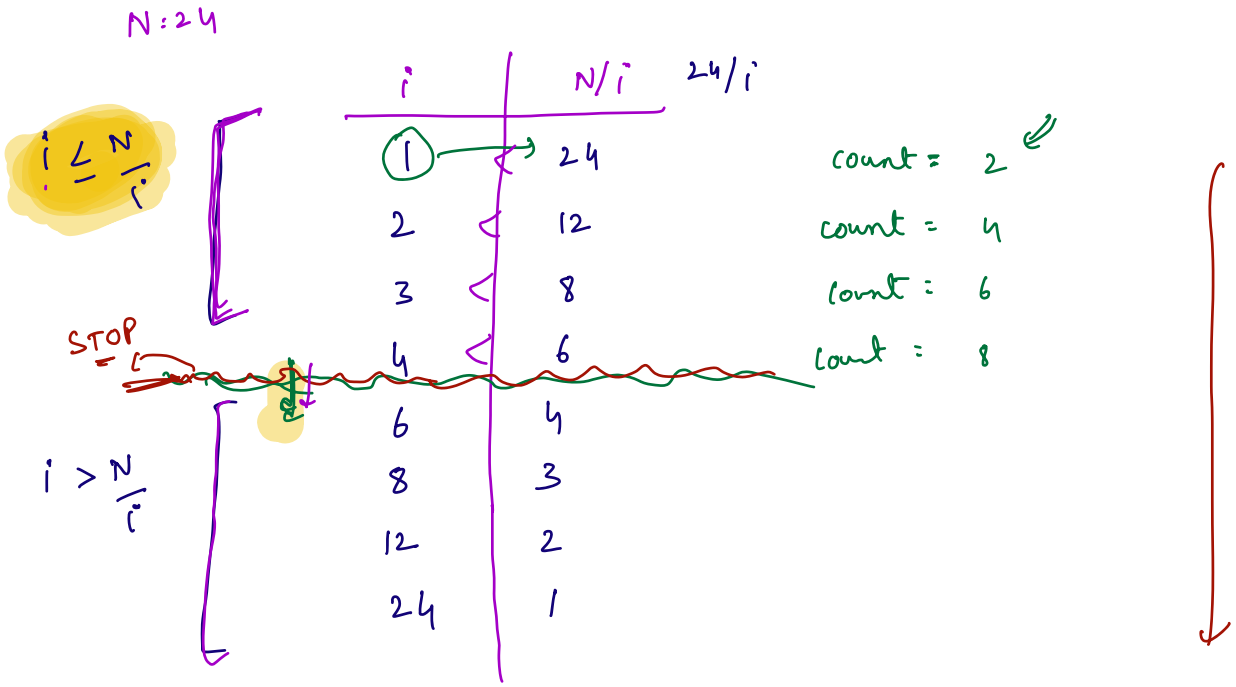
3 is a factor of 24

$\frac{24}{3} = 8$ remainder = 0, quotient = 8

4 is a factor of 24 $\Rightarrow$

$\frac{24}{4} = 6$

$\Rightarrow$ If i is a factor of N, $\frac{N}{i}$ is also a factor



$i \le \frac{N}{i}$

$\Rightarrow$ $i \times i \le N$

$i \times i \le \frac{N}{i} \times i$

$\Rightarrow$ $i \times i \le N$

```
int count = 0;
for( i = 1; i x i <= N; i++){
    if( N % i == 0){
        count = count + 2;
    }
}
```

$\Rightarrow$ 1, 2, 4, 5, 10, 20, 25, 50, 100 $\Rightarrow$ 9

$N = 100$

$\Downarrow$ 10 iterations

$i \times i \leq 100$

i	$100/i$
1	100
2	50
4	25
5	20
10	10

$i == \frac{N}{i}$

count = 2
count = 4
count = 6
count = 8
count = 9

```
int count = 0;
for( i = 1; i*i <= N; i++){
    if( N % i == 0){
        if( i == N/i ){
            count = count + 1;
        }
        else {
            count = count + 2;
        }
    }
}
```

$i = 1$

$i = [1, \sqrt{N}]$

Range

$i \times i \leq N$

$\sqrt{i^2} \leq \sqrt{N}$

$i \leq \sqrt{N}$

$Max(i) = \sqrt{N}$

#iterations = $\sqrt{N}$

$N = 10^{18} \Rightarrow$

#iters = $\sqrt{N} = \sqrt{10^{18}} = 10^9$ iterations

⇓

10 seconds

Prime Numbers

2, 3, 5, 7, 11, 13, 17, ...

A number which is divisible only by 1 and itself.

if factors(N) == 2 => Prime Number

```
boolean isPrime(int N) {
    if( factors(N) == 2) {
        return true;
    }
    else{
        return false;
    }
}
```

8:33 am

Question:

Given an array of size N, reverse the array [without any extra array]

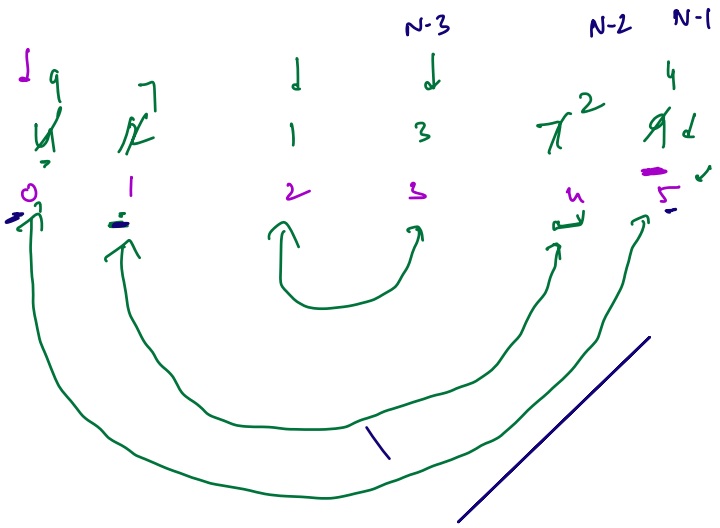
N = 5

A = [2, -1, 0, 4, 6]
0 1 2 3 4

[0, N-1]
[0, 4]

revCA) : [6 4 0 -1 2]
0 1 2 3 4

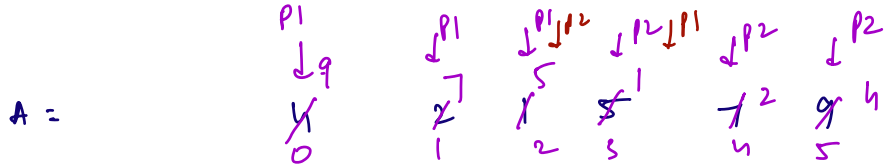
A :



0 ⟷ N-1

1 ⟷ N-2

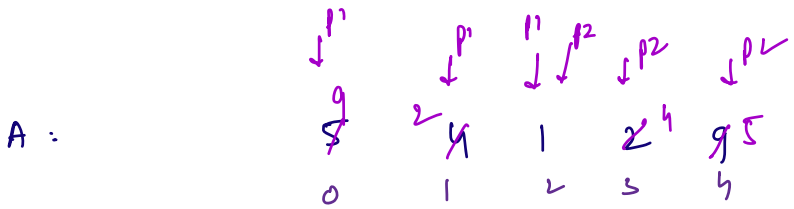
2 ⟷ N-3



p1 = 0; p2 = N-1

p1++ ; p2--

p1 < p2 ✓



p1 == p2

```
void reverse (int[] A, int N){

    int p1 = 0;
    int p2 = N-1;

    while( p1 < p2 ){
        int temp = A[p1];
        A[p1] = A[p2];
        A[p2] = temp;
        p1++; p2--;
    }
    return;
}
```

Question: Revers a part of an array

Ex1: A: [4, 5, 1, 2, 9, 6, 3, 7]
0 1 2 3 4 5 6 7

S = 2
E = 5

A': [4 5 6 9 2 1 3 7]
0 1 2 3 4 5 6 7

S E

Ex2: A: [-3, 4, 6, 9, 7, 8, 7, 2, 10]
0 1 2 3 4 5 6 7 8

S = 2, E = 6

A: [-3, 4, 7, 8, 7, 9, 6 2, 10]

N = 10

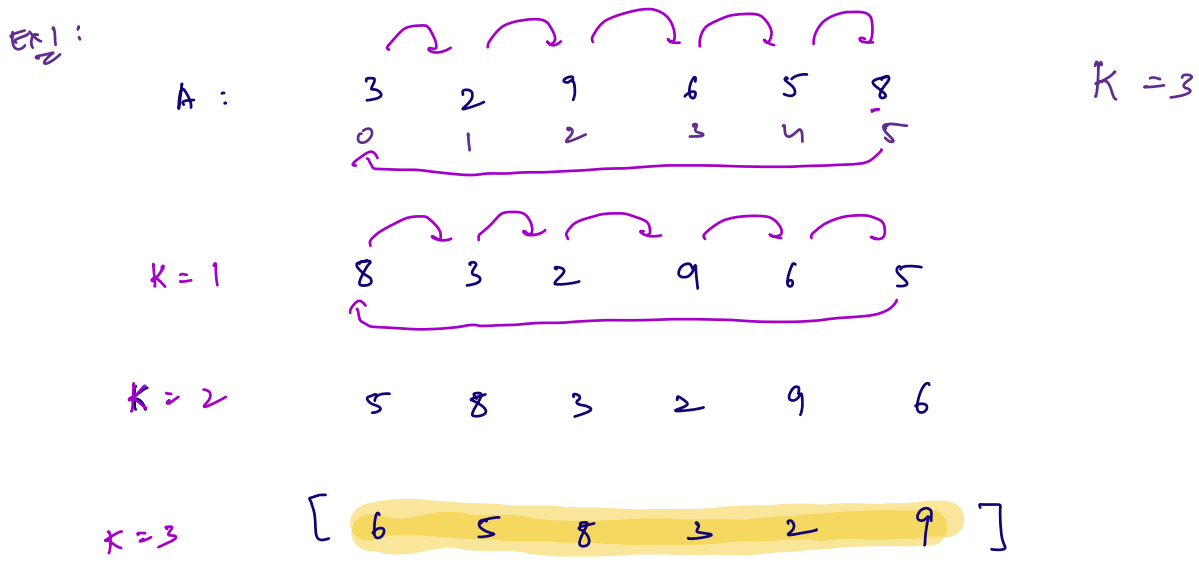
E = 1000

```
void reverse (int[] A, int N, int S, int E){
    int p1 = S;        A[S]
    int p2 = E;        A[E]

    while( p1 < p2 ){
        int temp = A[p1];
        A[p1] = A[p2];
        A[p2] = temp;
        p1++; p2--;
    }
    return;
}
```

Question: Given an array and integer K, right rotate the array K times (< N)

Ex1:



Ex2:

A = [9 5 4 3 2 7]

K = 1 7 9 5 4 3 2

K = 2 2 7 9 5 4 3

K = 3 3 2 7 9 5 4

K = 4 4 3 2 7 9 5

Approach 1:

Array Pg 2 (Support)

```
for (i = 0; i < k; i++){
    rotateArray(A);
}
```

→ N iterations

↘ [You should implement]

#iter : N × K ⇒ N × K

Approach 2:

A : 3 (0) −2 (1) 1 (2) 4 (3) 6 (4) 9 (5) 8 (6)

R_1 : 8 3 −2 1 4 6 9

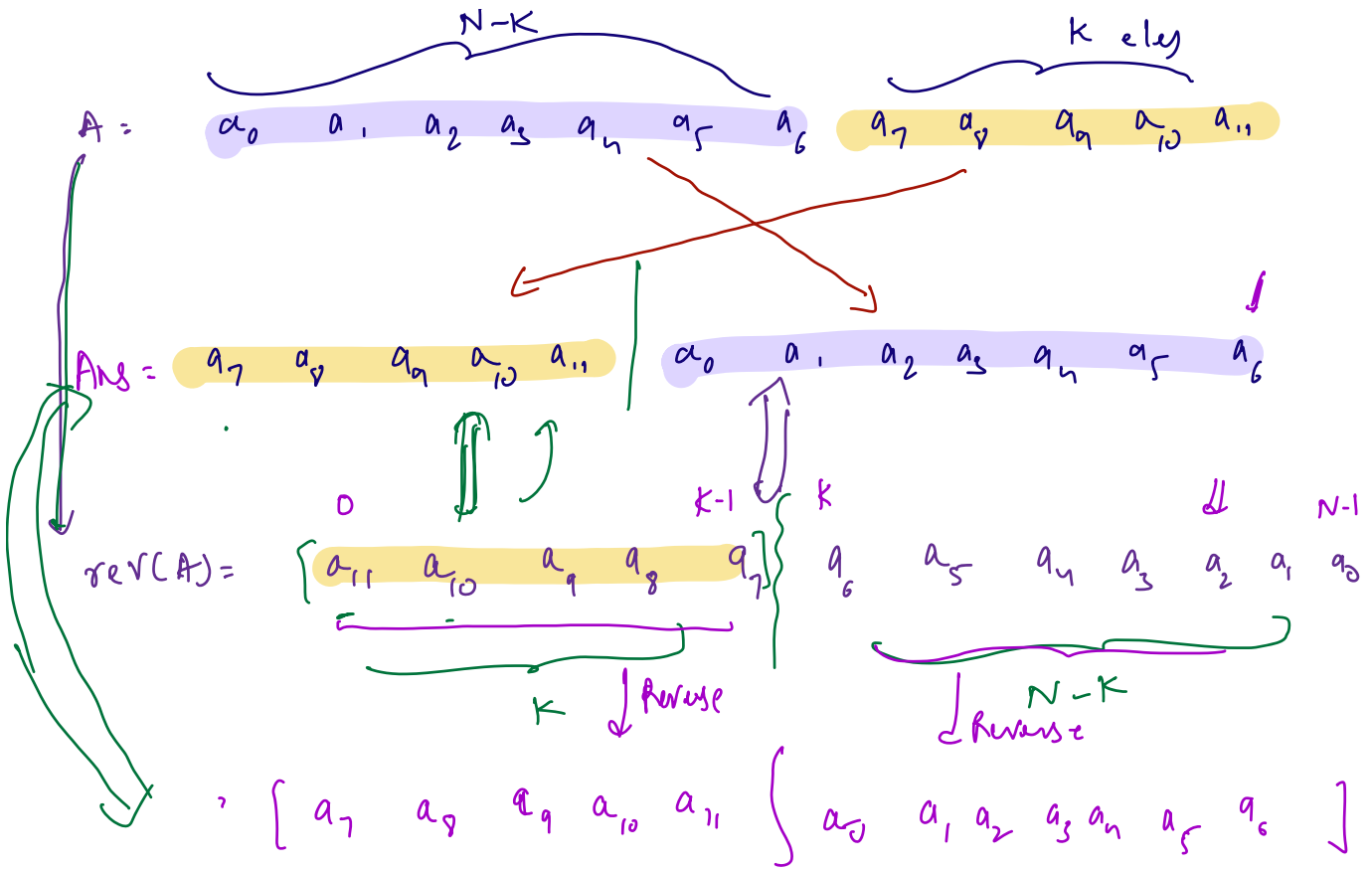
R_2 : 9 8 3 −2 1 4 6

R_3 : 6 9 8 3 −2 1 4

R_4 : 4 6 9 8 3 −2 1

R_5 : 1 4 6 9 8 3 −2

Observation: When we rotate array by K times, last K elements of original array will come to the front



Steps

1) Reverse Entire array => reverse(A, 0, N-1)

2) Reverse the first k eles => reverse(A, 0, K-1);

3) Reverse the remaining N-K eles
=> reverse(A, K, N-1);

A = 4 5 6 1 2 3 9 7

0 1 2 3 4 5 6 7

K = 4 => [0, 4-1]

```
void rotateArray (int[] A, int N, int K) {

    reverse (A, 0, N-1);

    reverse (A, 0, K-1);

    reverse (A, K, N-1);

    return;
}
```

Question: $K > N$

$K = 987$

$N = 5$

$A = a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4$

$R_1 = a_4 \quad a_0 \quad a_1 \quad a_2 \quad a_3$

$R_2 = a_3 \quad a_4 \quad a_0 \quad a_1 \quad a_2$

$R_3 = a_2 \quad a_3 \quad a_4 \quad a_0 \quad a_1$

$R_4 = a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_0$

$R_5 = a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4$

$R_6 = a_4 \quad a_0 \quad a_1 \quad a_2 \quad a_3$

$R_7 = a_3 \quad a_4 \quad a_0 \quad a_1 \quad a_2$

$K = 0, 5, 10, 15 \ldots \Rightarrow K\%5 = 0 \Rightarrow$ 0 times

$K = 1, 6, 11, 16 \ldots \Rightarrow K\%5 = 1 \Rightarrow$ 1 time

$K = 2, 7, 12, 17 \ldots \Rightarrow K\%5 = 2 \Rightarrow$ 2 times

$K = 3, 8, 13, 18 \ldots \Rightarrow K\%5 = 3 \Rightarrow$ 3 times

$K = 4, 9, 14, 19 \ldots \Rightarrow K\%5 = 4 \Rightarrow$ 4 times

$K = K \% N$

N: size of array

=> Lots of code

=> Errors / Wrong Answer / TLE errors

Steps to Solve a problem

-> Read the Question properly

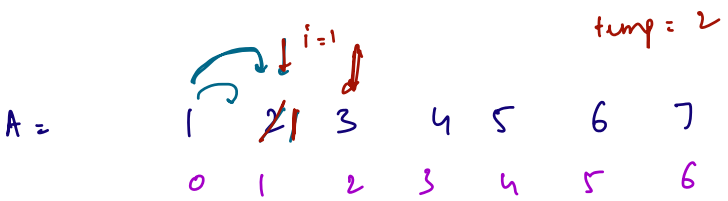
=> Sample Inputs & Output

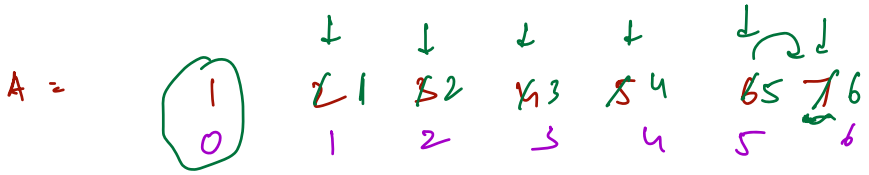
=> Take your own examples and see what should be the answers

-> Come up with a Brute Force (Naive Approach)

=> -> Make Observations

-> Optimize the code





temp = A[6]

```
for(i = N-1; i >= 1 ; i--){
    A[i] = A[i-1];
}
```

A[0] = temp;

```
temp1: A[N-1];
for(i: N-1; i ≥ 1; i--){
        A[i] = A[i-1];
}
A[0] = temp;
```

30-60 mins | Hints | Discussion Form/FA