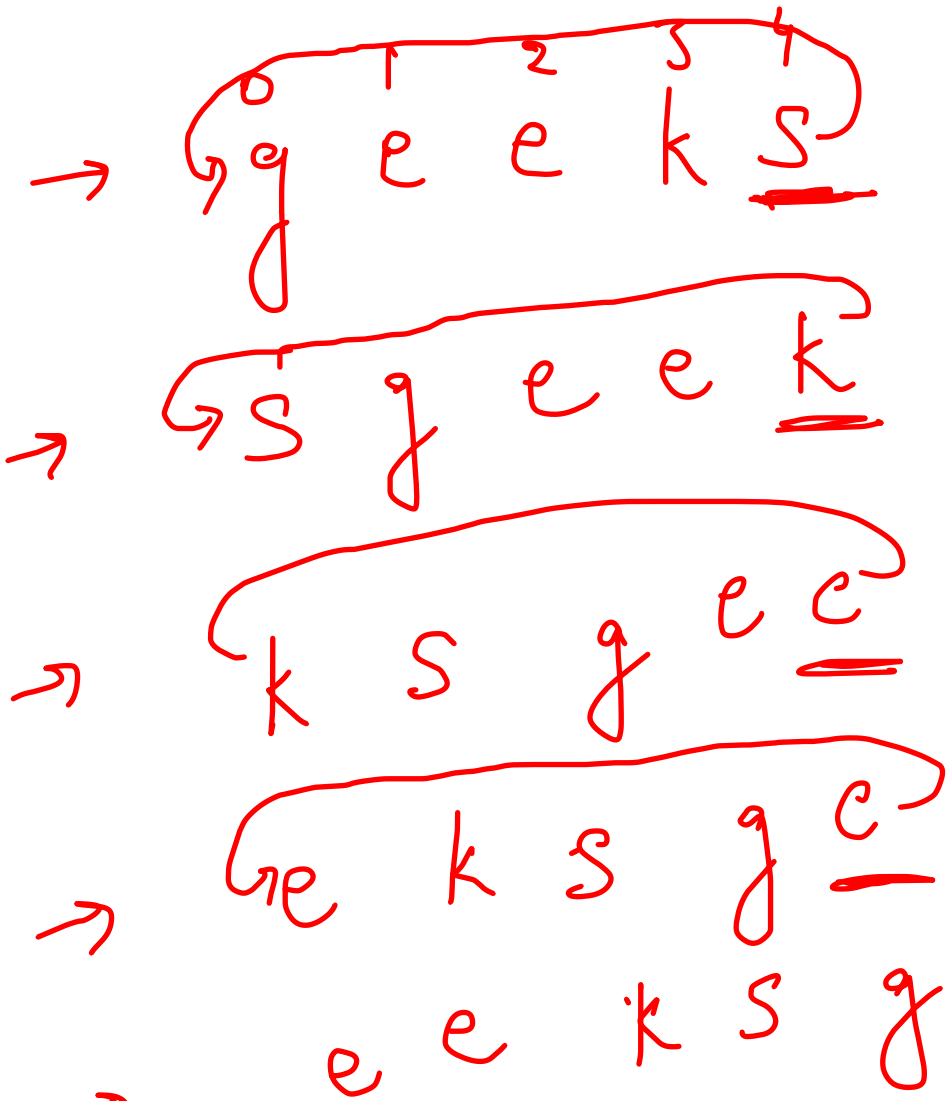
Str = "geeks";



substring (str.length() - 1) → s

+

substring (0, str.length() - 1) → geek

o/p → sgeek ✓

```
        Scanner s = new Scanner(System.in);
        String str = s.nextLine();
        int length = str.length();

        // print all the rotations

        String rotateString = str;
        for(int i = 0; i < length; i++){
            System.out.println(rotateString);
            rotateString = rotate(rotateString);
        }
    }

    public static String rotate(String str){
        int length = str.length();
        return str.substring(length - 1) + str.substring(0, length - 1);
    }
```

str = "geeks"

len = 5

RS = "geeks"

S + geek = Sgeek

k + Sgee = ksgee

O/P → geeks

→ Sgeek

→ ksgee

→ eksge

→ eeksg

```java
public static String findFirst(String words[], int n){
    for(int i = 0; i < n; i++){
        if(isPalindrome(words[i])){
            return words[i];
        }
    }
    return "empty";
}

public static boolean isPalindrome(String word){
    int left = 0;
    int right = word.length() - 1;

    while(left < right){
        if(word.charAt(left) != word.charAt(right)){
            return false;
        }
        left++;
        right--;
    }
    return true;
}
```

```java
Scanner s = new Scanner(System.in);
int n = s.nextInt();
s.nextLine(); // continue to new line character

// string array

String words[] = new String[n];
for(int i = 0; i < n; i++){
    words[i] = s.next();
}

String result = findFirst(words, n);
System.out.println(result);

}
```

Tc = O(n)

n = 2

{ def , ghi }
0 1

o/p → empty

i = 0, def
0 1 2
l → r
d. != f T
0

i = 1, ghi
0 1 2
f → r
g != e T

i = 2 X

str = "abcd"

Target = "cdab" ✓

Target = "dacb" — F

str + str

abcdabcd

abcdabcd

abcdabcd

str = "wxyz" → 4

Target = "xywz" → 4

concatstr = wxyzwxyz

return false.

o/p = false.

```
    Scanner s = new Scanner(System.in);
    String str = s.nextLine();
    String target = s.nextLine();

    boolean result = rotation(str, target);
    System.out.println(result ? "True" : "False");
}

public static boolean rotation(String str, String target){
    if(str.length() != target.length()){
        return false;
    }

    String concatStr = str + str;
    return concatStr.contains(target);
}
```

TC → O(n), SC → O(1)

four Sum-

found = false.

0 1 2 3 4 5 6 7

1 1 2 2 3 3 5 5

```
for(int i=0; i<n-3; i++)
    //skip dup.
    if(i>0 && arr[i]==[i+1]
        continue;

for(j=i+1; i<n-2; j++){
    //skip dup.
if(j>i+1 && arr[j] {j+1)
        continue;
```

```
int l = j+1
int r = n-1
while (l<r)
long sum = arr[i] + [j] + [l] + [r]
if (sum = target){
    found. true;
sysa([i] + [j] + [l] + [r])

while (l<r && [l] = [l+1]) l++
while ([l<r && [r]==[r-1] r--
    l++; r--;
else if (sum < target) l++
else r--
```