Given a string, we must count how many palindromic substrings it contains

Given a string aaa, there are 3 palindromic substrings: a, aa, aaa

The expected output should print the number of substrings within a string

If the given string is at least of length 2, then we check to see if both the characters are the same or not

The string from the main will be passed into the method countSubstrings

There will be an integer declared to hold the length of the string.

Integer answer will be set to 0, this will return the number of substrings

The center of a palindrome can be either position of 2N-1, so I will create a for loop starting from the center at 0, continuing until 2\*N-1

Within the forloop, the left interval will be divided by 2, the right will be the left+the centermod2

A while loop will run as long as the left is greater than or equal to 0, the right is less than N, and while the character left in string S is equal to the right. Within the while loop the answer will update 1, the left will decrease 1 (because we are advancing through the string), and right will increase 1.

After the for loop is executed the answer will be returned

int countSubstrings(String s){

int n = s.length(), answer = 0;

for (int center = 0; center <= 2\*n-1; ++center) {

int l = center / 2;

int r = l + center % 2;

while (l >= 0 && r < n && s.charAt(l) == s.charAt(r)) {

answer++;

l--;

r++;

}

}

return answer;

}