```
def custom_count(string, substring, allow_overlap):
    count = 0
    sub_len = len(substring)
    str_len = len(string)

    i = 0
    while i <= str_len - sub_len:
        # If the substring matches the slice of the string, increment the count
        if string[i:i + sub_len] == substring:
            count += 1
            # If overlapping is allowed, move by 1 step; otherwise, move by the length of the substring
            if allow_overlap:
                i += 1
            else:
                i += sub_len
        else:
            i += 1

    return count

# Example usages:

# Test case 1: Regular case without overlap
a = "sggs"
result = custom_count(a, 'gg', False)
print(result)  # Output should be 1

# Test case 2: Overlapping substrings allowed
b = "aaaa"
result = custom_count(b, 'aa', True)
print(result)  # Output should be 3 (matches: "aa", "aa", "aa")
```

```
# Test case 3: Overlapping substrings not allowed
result = custom_count(b, 'aa', False)
print(result)  # Output should be 2 (matches: "aa", "aa")

# Test case 4: Substring not found
c = "hello"
result = custom_count(c, 'z', False)
print(result)  # Output should be 0

# Test case 5: Empty substring
d = "hello"
result = custom_count(d, '', False)
print(result)  # Output should be 6 (one match at each character plus one at the end)

# Test case 6: Substring longer than string
e = "hi"
result = custom_count(e, 'hello', False)
print(result)  # Output should be 0

# Test case 7: Exact match
f = "hello"
result = custom_count(f, 'hello', False)
print(result)  # Output should be 1
```

output :

```
1
3
2
0
```