# string_manual_inbuilt_fun & method

## String-related functions without using inbuilt functions

```
def string_length(s):
    count = 0
    for _ in s:
        count += 1
    return count
```

```
def count_char(s, target):
    count = 0
    for ch in s:
        if ch == target:
            count += 1
    return count
```

```
def reverse_string(s):
    reversed_s = ''
    for i in range(string_length(s) - 1, -1, -1):
        reversed_s += s[i]
    return reversed_s

# 2nd method
def reverse_string(s):
    reversed_s = ''
    for i in s:
        reversed_s = i+reversed_s
    return reversed_s
```

```
def is_palindrome(s):
    n = string_length(s)
    for i in range(n // 2):
        if s[i] != s[n - i - 1]:
            return False
    return True
```

```
def to_lower_case(s):
    result = ''
    for ch in s:
        if 'A' <= ch <= 'Z':
            result += chr(ord(ch) + 32)
```

```
        else:
            result += ch
    return result
```

```
def to_upper_case(s):
    result = ''
    for ch in s:
        if 'a' <= ch <= 'z':
            result += chr(ord(ch) - 32)
        else:
            result += ch
    return result
```

```
def remove_spaces(s):
    result = ''
    for ch in s:
        if ch != ' ':
            result += ch
    return result
```

```
def count_words(s):
    count = 0
    in_word = False
    for ch in s:
        if ch != ' ' and not in_word:
            count += 1
            in_word = True
        elif ch == ' ':
            in_word = False
    return count
```

```
def replace_char(s, old, new):
    result = ''
    for ch in s:
        if ch == old:
            result += new
        else:
            result += ch
    return result
```

```
def find_substring(s, sub):
    n = string_length(s)
    m = string_length(sub)
    for i in range(n - m + 1):
        match = True
```

```
        for j in range(m):
            if s[i + j] != sub[j]:
                match = False
                break
        if match:
            return i
    return -1
```

```
def remove_vowels(s):
    result = ''
    for ch in s:
        if ch not in 'aeiouAEIOU':
            result += ch
    return result
```

```
def get_unique_characters(s):
    result = ''
    for ch in s:
        exists = False
        for r in result:
            if ch == r:
                exists = True
                break
        if not exists:
            result += ch
    return result
```

```
def char_frequency(s):
    freq = {}
    for ch in s:
        if ch in freq:
            freq[ch] += 1
        else:
            freq[ch] = 1
    return freq
```

```
def capitalize_first_letter(s):
    result = ''
    capitalize = True
    for ch in s:
        if ch == ' ':
            result += ch
            capitalize = True
        elif capitalize and 'a' <= ch <= 'z':
            result += chr(ord(ch) - 32)
```

```
            capitalize = False
        else:
            result += ch
            capitalize = False
    return result
```