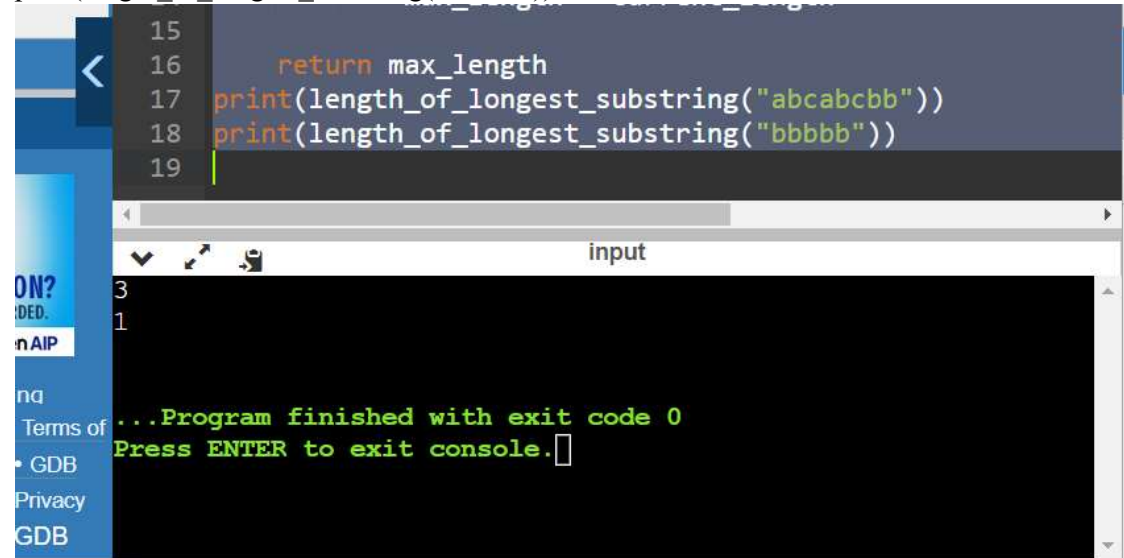


Problem 6: Given a String, find the length of longest substring without any repeating character.

```
def length_of_longest_substring(s):  
    max_length = 0  
    char_map = {}  
    start = 0  
  
    for end in range(len(s)):  
        if s[end] in char_map and char_map[s[end]] >= start:  
            start = char_map[s[end]] + 1  
  
        char_map[s[end]] = end  
        current_length = end - start + 1  
  
        if current_length > max_length:  
            max_length = current_length  
  
    return max_length  
print(length_of_longest_substring("abcabcbb"))
```

```
print(length_of_longest_substring("bbbbbb"))
```



The image shows a code editor window with a dark theme. The code is written in Python and includes a function definition and two print statements. The line numbers 15 through 19 are visible on the left. Below the code editor is a terminal window with a black background and green text. The terminal shows the output of the program, which is '3' and '1', followed by a message indicating the program finished with exit code 0 and a prompt to press ENTER to exit the console.

```
15  
16     return max_length  
17 print(length_of_longest_substring("abcabcbb"))  
18 print(length_of_longest_substring("bbbbbb"))  
19
```

input

3
1

...Program finished with exit code 0
Press ENTER to exit console.