```python
#Write a function that takes a string and returns the length of the longest substring without
#repeating characters.
#• Input: A string s.
#• Output: An integer representing the length of the longest substring without repeating
#characters.

'''s = "abcabcbb"
def len_subString(s):
    result_set = set()
    result = 0
    l = 0
    for i in range (0,len(s)):
        while s[i] in result_set:
            result_set.remove(s[l])
            l+=1
        result_set.add(s[i])
        result = max(result,i-l+1)

    print(result)

len_subString(s)'''



#Write a function to determine if two strings are anagrams of each other. Two strings are anagrams if
#they contain the same characters with the same frequencies but in any order.
#• Input: Two strings s and t.
#• Output: A boolean value (True or False) indicating if the strings are anagrams.
#Hint: Use a counter or hash map to count the frequency of characters in both strings and compare them.

'''s = "cinema"
t = "iceman"

def anagram(s,t):
    if (sorted(s)) == (sorted(t)):
        print(f"True - The strings \"{s}\" and \"{t}\" are anagrams.")
    else:
        print(f"False - The string \"{s}\" and \"{t}\" are not anagrams")

anagram(s,t)'''

#or

'''s = "cinema"
t = "iceman"
x =[s[i] for i in range (0,len(s))]
x.sort()
y = [t[i] for i in range (0,len(t))]
```

```
y.sort()
if (x == y):print("The strings are anagrams")
else: print("The strings aren't anagrams.")
'''

#or

'''from collections import Counter
s = "cinema"
t = "iceman"

def anagram(s,t):
    if (Counter(s)) == (Counter(t)):
        print(f"True - The strings \"{s}\" and \"{t}\" are anagrams.")
    else:
        print(f"False - The string \"{s}\" and \"{t}\" are not anagrams")
anagram(s,t)'''
```