

```

1 #Write a function that takes a string and returns the length of the longest
  substring without
2 #repeating characters.
3 #• Input: A string s.
4 #• Output: An integer representing the length of the longest substring without
  repeating
5 #characters.
6
7 '''s = "abcabcbb"
8 def len_subString(s):
9     result_set = set()
10    result = 0
11    l = 0
12    for i in range (0,len(s)):
13        while s[i] in result_set:
14            result_set.remove(s[l])
15            l+=1
16        result_set.add(s[i])
17        result = max(result,i-l+1)
18
19    print(result)
20
21 len_subString(s)'''
22
23
24
25 #Write a function to determine if two strings are anagrams of each other. Two
  strings are anagrams if
26 #they contain the same characters with the same frequencies but in any order.
27 #• Input: Two strings s and t.
28 #• Output: A boolean value (True or False) indicating if the strings are
  anagrams.
29 #Hint: Use a counter or hash map to count the frequency of characters in both
  strings and compare them.
30
31 '''s = "cinema"
32 t = "iceman"
33
34 def anagram(s,t):
35     if (sorted(s) == (sorted(t))):
36         print(f"True - The strings \"{s}\" and \"{t}\" are anagrams.")
37     else:
38         print(f"False - The string \"{s}\" and \"{t}\" are not anagrams")
39
40 anagram(s,t)'''
41
42 #or
43
44 '''s = "cinema"
45 t = "iceman"
46 x =[s[i] for i in range (0,len(s))]
47 x.sort()
48 y = [t[i] for i in range (0,len(t))]

```

```
49 y.sort()
50 if (x == y):print("The strings are anagrams")
51 else: print("The strings aren't anagrams.")
52 '''
53
54 #or
55
56 '''from collections import Counter
57 s = "cinema"
58 t = "iceman"
59
60 def anagram(s,t):
61     if (Counter(s)) == (Counter(t)):
62         print(f"True - The strings \"{s}\" and \"{t}\" are anagrams.")
63     else:
64         print(f"False - The string \"{s}\" and \"{t}\" are not anagrams")
65 anagram(s,t)'''
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```