**Assignment 1:**

Given a string **s** and a number **x**, print the shortest substrings which start and end with the same character and have lengths greater than or equal to **x**. If multiple substrings exist with the same shortest length, print them all.

**Answer:**

Def print\_shortest\_substrings(s, x):

substrings=[]

for i in range(lens(s)):

for j in range(i+x-1, len(s)):

if s[i] = = s[j]:

substring = s[i:j+1]

if len(substring)>=x and (not substrings or len

(substring) = = len(substrings[0])):

substrings.append(substring)

elif len(substring) < len(substrings[0]):

substrings = [substring]

if substrings:

for substrings in substrings:

print(substrings)

else:

print(“not-found”)

# Example usage

s = "abccdbacca"

x = 3

print("x =", x)

print\_shortest\_substrings(s, x)

x = 4

print("\nx =", x)

print\_shortest\_substrings(s, x)

x = 5

print("\nx =", x)

print\_shortest\_substrings(s, x)

x = 6

print("\nx =", x)

print\_shortest\_substrings(s, x)

x = 7

print("\nx =", x)

print\_shortest\_substrings(s, x)

x = 8

print("\nx =", x)

print\_shortest\_substrings(s, x)

**Assignment 2:**

Given a string **s**, find the ASCII value of each character iteratively. If the ASCII value is even, increment the next character by (**ASCII\_value % 7**). If the ascii value is odd, decrement the previous character by (**ASCII\_value % 5**). Output the newly formed string.

Note:

* If a character has already been changed once, do not change that character again.
* If the new number is an invalid ASCII value, **replace it with 83**.

**Answer:**

def transform\_string(s):

result = []

for i in range(len(s)):

ascii\_val = ord(s[i])

if i > 0 and (ascii\_val % 2 == 0):

new\_ascii = ascii\_val + (ascii\_val % 7)

result.append(chr(new\_ascii))

elif i < len(s) - 1 and (ascii\_val % 2 == 1):

new\_ascii = ascii\_val - (ascii\_val % 5)

result[-1] = chr(new\_ascii) if new\_ascii >= 0 and new\_ascii <= 127 else chr(83)

else:

result.append(s[i])

return ''.join(result)

# Example usage

s = "sHQen}"

result = transform\_string(s)

print("Original string:", s)

print("Transformed string:", result)