'''

Problem:

Given a string s, find the longest palindromic substring in s.

Requires:

if there exists multiple longest palindromics

the code should be efficient

Way:

for python, when check the palindromic, use s[::-1] to inverse the string

loop from long to short

'''

def is\_palindromic(s):

s\_inv = s[::-1]

if s == s\_inv:

return True

else:

return False

def search\_longest\_palin(str\_in):

num\_char = len(str\_in)

for len\_tar in range(num\_char, 0, -1):

Flag = False

result = []

for i in range(0, num\_char-len\_tar+1):

for j in range(i+len\_tar, num\_char):

if is\_palindromic(str\_in[i:j]):

result.append(str\_in[i:j])

Flag = True

if Flag:

return result

return None

str\_in = 'abcdeffedcabbcdeffedcb'

result = search\_longest\_palin(str\_in)

print(result)