'''

Problem:

Given a text txt[0..n-1] and a pattern pat[0..m-1] (n > m),

write a function search(char pat[], char txt[]) that prints all occurrences of pat[] in txt[].

Requires:

Use the KMP (Knuth Morris Pratt) Pattern Searching algorithm

Ways:

The core idea of KMP algorithm is:

whenever we detect a mismatch (after some matches),

we already know some of the characters in the text of next window.

We take advantage of this information to avoid matching the characters that we know will anyway match

Waiting:

The implementation of calculating the next array in a elegant way is incorrect

'''

def get\_next\_array(arr):

value = 1

# for ilen in range(1, len(arr)-1):

for ilen in range(1, len(arr)):

fore\_prex = arr[:ilen]

back\_prex = arr[len(arr)-ilen:]

if fore\_prex == back\_prex:

value = ilen + 1

return value

def get\_next(array):

next\_array = [0] \* len(array)

for iarr in range(1, len(array)):

next\_array = get\_next\_array(array[:iarr])

return next\_array

str\_pri = ‘abcabdabeabfabg’

str\_tar = ‘abeabf’

next\_val = get\_next(str\_tar)

jtar = 0

while ipri < len(str\_pri):

if str\_pri[ipri] == str\_tar[jtar]:

if jtar == len(str\_tar) - 1:

print(‘match at the place %d : %d’ % (ipri - len(str\_tar) + 1, ipri))

jtar = 0

else:

ipri += 1

jtar += 1

continue

else:

# add a line here

ipri += 1

jtar = next\_val[jtar]