def computeLPSArray(pat, M, lps):

    len **=** 0  *# length of the previous longest prefix suffix*

    lps[0]  *# lps[0] is always 0*

    i **=** 1

*# the loop calculates lps[i] for i=1 to M-1*

**while** i **<** M:

**if** pat[i] **==** pat[len]:

            print(f'Match pat[{i=}]==pat[{len=}] and store and forward')

            len **+=** 1

            lps[i] **=** len

            i **+=** 1

            print(f'Len incremented {len=}, lps[{i=}]={len=}, i incremented {i=}')

**else**:

*# This is tricky. Consider the example.*

*# AAACAAAA and i = 7. The idea is similar to search step.*

**if** len **!=** 0:

                len **=** lps[len**-**1]

                print(f'Mismatch rollback len=lps[len-1]: {len=}')

*# Also, note that we do not increment i here*

**else**:

                print(f'Mismatch rollback lps[{i=}]=0, increment i {i=}')

                lps[i] **=** 0

                i **+=** 1

pat **=** 'AABAACAABAA'

01234567890

Match pat[i=1]==pat[len=0] and store and forward

Len incremented len=1, lps[i=2]=len=1, i incremented i=2

Mismatch rollback len=lps[len-1]: len=0

Mismatch rollback lps[i=2]=0, increment i i=2

Match pat[i=3]==pat[len=0] and store and forward

Len incremented len=1, lps[i=4]=len=1, i incremented i=4

Match pat[i=4]==pat[len=1] and store and forward

Len incremented len=2, lps[i=5]=len=2, i incremented i=5

Mismatch rollback len=lps[len-1]: len=1

Mismatch rollback len=lps[len-1]: len=0

Mismatch rollback lps[i=5]=0, increment i i=5

Match pat[i=6]==pat[len=0] and store and forward

Len incremented len=1, lps[i=7]=len=1, i incremented i=7

Match pat[i=7]==pat[len=1] and store and forward

Len incremented len=2, lps[i=8]=len=2, i incremented i=8

Match pat[i=8]==pat[len=2] and store and forward

Len incremented len=3, lps[i=9]=len=3, i incremented i=9

Match pat[i=9]==pat[len=3] and store and forward

Len incremented len=4, lps[i=10]=len=4, i incremented i=10

Match pat[i=10]==pat[len=4] and store and forward

Len incremented len=5, lps[i=11]=len=5, i incremented i=11

[0, 1, 0, 1, 2, 0, 1, 2, 3, 4, 5]