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=}]')

            len **+=** 1

            print(f'Len incremented {len=}, lps[{i=}]=len, ++i={i**+**1}')

            lps[i] **=** len

            i **+=** 1

**else**:

*# This is tricky. Consider the example AAACAAAA and i=7.*

*# The idea is similar to search step.*

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

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

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

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

**else**:

                print(f'Mismatch reset lps[{i=}]=0, ++i={i**+**1}')

                lps[i] **=** 0

                i **+=** 1

pat **=** 'AABAACAABAA'

01234567890

Match pat[i=1]==pat[len=0]

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

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

Mismatch reset lps[i=2]=0, ++i=3

Match pat[i=3]==pat[len=0]

Len incremented len=1, lps[i=3]=len, ++i=4

Match pat[i=4]==pat[len=1]

Len incremented len=2, lps[i=4]=len, ++i=5

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

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

Mismatch reset lps[i=5]=0, ++i=6

Match pat[i=6]==pat[len=0]

Len incremented len=1, lps[i=6]=len, ++i=7

Match pat[i=7]==pat[len=1]

Len incremented len=2, lps[i=7]=len, ++i=8

Match pat[i=8]==pat[len=2]

Len incremented len=3, lps[i=8]=len, ++i=9

Match pat[i=9]==pat[len=3]

Len incremented len=4, lps[i=9]=len, ++i=10

Match pat[i=10]==pat[len=4]

Len incremented len=5, lps[i=10]=len, ++i=11

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