NAME: JONATHAN AKASH PATTA
ASU ID: 1229738616

Q2)

To prove that X and Y are from the same circular sequence we can take the new string and concatenate it with itself to produce all possible rotations of X. if Y is a substring of the new string then this proves that Y is a part of the same circular sequence. If not we can repeat the same thing for Y and try again.

1) Concatenate X with itself to create a new sequence X' = XX. This effectively creates all possible rotations of X.
2) Use the Z-algorithm to search for Y within X':
   Construct the concatenated string S = Y + "$" + X', where "$" is a special character not present in X or Y.
   Compute the Z-array for S.
3) If there exists an index i in the Z-array where Z[i] = |Y| (the length of Y), and i > |Y| + 1, then Y is a substring of X'.
   This means Y matches a rotation of X, indicating X and Y come from the same circular sequence.

4) If no such match is found, repeat steps 1-3 with X and Y swapped (i.e., search for X in Y'=YY).

Q3)

Original lps: [0, 0, 0, 1, 2, 3, 0]
Modified lps': [0, 0, 0, 0, 0, 3, 0]

Original KMP Algorithm:
i = 0, j = 0: T!=A
i = 1, j = 0: C!=A
I = 2, j = 0: A=A
I = 3, j = 1: T=T
i = 4, j = 2: C=C
i = 5, j = 3: A=A
i = 6, j = 4: T=T
i = 7, j = 5: G!=C
i = 7, j = 2: G!=C
i = 7, j = 0: G!=A
i = 8, j = 0: A=A

i = 9, j = 1: T=T
i = 10, j = 2: G!=C
i = 10, j = 0: G!=A
i = 11, j = 0: A=A
i = 12, j = 1: T=T
i = 13, j = 2: C=C
i = 14, j = 3: A=A
i = 15, j = 4: G!=A
i = 16, j = 5: C=C
i = 17, j = 6: T=T

Modified KMP Algorithm (using lps'):
i = 0, j = 0: T!=A
i = 1, j = 0: C!=A
I = 2, j = 0: A=A
I = 3, j = 1: T=T
i = 4, j = 2: C=C
i = 5, j = 3: A=A
i = 6, j = 4: T=T
i = 7, j = 5: G!=C
i = 7, j = 0: G!=A
i = 8, j = 0: A=A
i = 9, j = 1: T=T
i = 10, j = 2: G!=C
i = 10, j = 0: G!=A
i = 11, j = 0: A=A
i = 12, j = 1: T=T
i = 13, j = 2: C=C
i = 14, j = 3: A=A
i = 15, j = 4: G!=A
i = 16, j = 5: C=C
i = 17, j = 6: T=T

The original kmp algorithm has 21 matches and the modified algorithm has 20 matches(The extra comparison is highlighted in yellow) . Both algorithms will correctly identify the pattern and although the second algorithm is faster the time gained using the modified algorithm is not very significant as it is still a linear time algorithm, O(n+m).

Q4)

Taking the previous example

Z array: [0, 0, 0, 3, 0, 0, 0]

lps': [0, 0, 0, 0, 0, 3, 0]

The main difference between Z array and lps' is that the length of the sequence is given at the end of the sequence hence it is simply enough to shift the value in the array by an offset equal to one less than the size of the prefix match.

Assumptions:

- the computed Zarray is valid(no need for boundary checks)

Pseudo code:

```
Lpsdash = [0,0,...]

ZarrayToLPSdash(Z[]):
        For i,element in Z array:
                Lpsdash[element+i-1]= Z[i]
```