Assignment 2 (9th August 2023)

Team Number: 23

Team Members:

Yash Kumar (21CS30059)

Sampreeth RS (21CS30038)

Algorithm: We have considered 3 arrays, 1st for permutation 1, 2nd for permutation 2 and 3rd for the final permutation formed out of both. If we consider array for 1st permutation to be a[], for 2nd to be b[] and for 3rd to be c[], then for any index i: c[i]=b[a[i]] and this is the logic we have used to firstly form the array c[].

Once we have the array c[], we iterate over all indices of c (0-9). If for any index i, c[i]=i, then it does not form a cycle and we mark it as visited. Secondly, if for any index i, c[i] is not equal to i, then we mark i as visited, print c[i] and make i=c[i]. (We also print the opening parenthesis) We keep doing this unless we come across a visited index, which means completion of a cycle. Once a cycle is identified we print the closing parenthesis. Then we again increment value of i from which we had started investigation of the cycle.

Time Complexity: O(2N)

Reason: c array can be built in O(N) time as we only do constant time operation on each element of c

Printing of cycles can be done in O(2N) time as each element of the array c is visited atmost twice (once during printing of cycle and once in the loop iteration)