**COA LAB 09-08-2023 Wednesday**

**Assignment Number :** 1 **Problem Number :** 1

**Group Number :** 9 **Semester :** AUTUMN 2023

**Group Members :**

* Meduri Harshith Chowdary (21CS10042)
* Maddi Nihith (21CS10040)

**Algorithm :**

* Firstly declare 3 integer arrays for left, right and output permutations.
* Then we take inputs of all cycles of left permutation, while taking input of all cycles if a number gets repeated i.e., if it appears in more than one cycle then the input permutation is wrong.
* Similarly for the right permutation as well.
* While taking cycles of left permutation as input we map left[cycle[i]] = cycle[i+1] for all i except the last one for which we store cycle[0].
* Similarly for the right permutation as well.
* Now we iterate over all elements of left permutation and map output[i] = right[left[i]] for each i from 0 to 9.
* Now we discard left permutation to mark whether each number is visited or not.
* Again we iterate over 0 to 9 and for each i we continue to output[i] until output[i] is equal to the original i which we store as j (say). Now if number of elements accessed in between is >1 this makes up a cycle or it has to be a self loop i.e., I would be equal to j which we do not print.
* Now we iterate to next i=1, check if it is already visited or not if already visited continue to i=2 and similarly. If not visited till now start a new cycle exploration from this i.