**Assignment 2, Question 1**

**Group 24**

**Gorantla Thoyajakshi – 21CS10026**

**Ashwin Prasanth – 21CS30009**

**Algorithm:**

* Make a hash table with entries corresponding to 0-9, initialised with hashtable[i]=i
* Go to the first permutation. For each entry in the hash table, do the following:
  + Go through each cycle, if the hash index (number itself) is located, then make hash table entry = the number next to it, unless it is the last number in the cycle, in which case, you make the first number of the cycle = hash table entry
  + If no match found in all cycles, then hash table entry = number itself
* Now go to the second permutation, and update the hash table as follows:
  + Go through each cycle, and look for the number stored (entry) in the cycle, and update hash table entry in the same way as mentioned above
  + If no match is found, then there is no change to the hash table entry
* **Now we have a hash table that will give you the permutation. We need to convert it into cycle notation. For this we do the following:** 
  + For each element in the hashtable, maintain a Boolean poc (part of cycle)
  + Make a variable ‘start’ and an array ‘visited’
  + Start from initial index (0), set start = index, append index to ‘visited’, set poc to 1
    - Set index = entry of hash table, and go to hashtable[index], and append it to ‘visited’, and set poc to 1
    - Repeat until hashtable[index] = start
  + Move to the next index with poc=0, and repeat
  + Stop when all elements have poc=1
* This algorithm has an approximate time complexity of O(N^2), and a space complexity of O(N) to store the hash table
* Since N is 10, O(N^2) would not be significant, hence we decided to proceed with this algorithm