Assignment 2 Solution

To solve the problem we created 3 different arrays of size 10 which initially stores the values from 0 to 9 .

The three different arrays are f,g,h which will store all values from 0 to 9 . Ex 0123456789. Size =10.

Now we ask the user for the number of cycles in permutation, which will be stored in s0 . Now we run loop1 s0 number of times,Each time we ask the user the size of the cycle he needs. Now we take variables which are next and prev , these variables store the values in the cycle array continuously. After storing next and prev , we do the according changes in the f array.

After changing f according to cycle 1 in permutation 1 we will proceed with the next cycle in permutation 1 .

This loop will continue till we change the initial f array , s0 no of times . After doing these changes we will get the final cycle representation matrix of permutation 1.

We will repeat the same process with the second permutation and get the final cycle representation matrix for it.

For the product of the two cycles we used the arrays f,g and stored the results in array h. For example let's take value a=1 now after doing first permutation cycles the value will be f(1) and after this going through the second permutation cycles , the value becomes g(f(1)) this value will be stored in array s which is required array representation of the product permutation.

Similarly we will do the above process for all the integers in the range 0 to 9 and get the final permutation array representation . Now we need to convert this array to cycle representation.

To convert this to cycle representation we can take visited array which contains values 0 for all i=0 to 9 , now we will traverse through the visited array and whenever
the visited is 0 we will store the value present at that index to temp and print this number , now we will run a loop similar to dfs till we get the temp value again ,when we get the temp value we break the loop and search for the newest value in the visited array which we haven't visited yet and repeat the process above .

Using the above process we will be able to print the final permutation array into the cycle representation.