Name: Pushpinder Singh
Reg No. 14MCS1036

Assignment: The passport office has one main counter and 5 sub counters for processing the applications. A maximum of 100 applicants can be handled by the main counter. Every applicant must visit the main counter then any one of the sub counters. All the applicants have application unique ID starting from 1 to 100. At the main counter all are given the same token ID. Based on their application ID they are sent to different subcounters (For example application ID 1 to 5 are sent to counters 1 to 5 and then ID 6 is sent to counter 1, ID 7 to counter 2 and so on). At the subcounters Applicants are checked if they have visited main counter and then given the token ID equal to that conuter ID. Write an application to simulate this scenario and for every applicant print the original main token ID and the next token ID that is assigned by the subcounters.

Program 1:
Filename: fprivate1.c
Output: Gives unsynchronized output based on whichever thread finishes first.

```c
#include<stdio.h>
#include<omp.h>

int get_app_id(int i) /*function to assign Application ID to customer*/
{
return(i);
}

int get_main_token(int app_id) /*function to assign Main Counter Token ID to customer
according to his Application ID*/
{
return(app_id);
}

int get_sub_token(int x) /* function to assign Sub counter token ID to a customer after he
has been processed by the main counter*/
{
int k=x%5;
if(k==0)      /*when main token ID is a multiple of 5*/
return(k+5); /*customer will go to counter 5 as there is no counter 0 */
else
return(k);
}



main()
{
int app_id=0,main_token=0,sub_token=0,i;

#pragma omp parallel for firstprivate(app_id,main_token,sub_token)
for(i=1;i<=100;i++)
  {
  app_id=get_app_id(i);     /*private int i is used to generate application ID's for
  customers*/
  main_token=get_main_token(app_id);
  sub_token=get_sub_token(main_token);

  printf("\nMain counter is currently processing Application ID:%d \nMain Counter Token ID
  :%d \nCounter ID assigned:%d",app_id,main_token,sub_token);

  }

}
```

```c
/*file name: fprivate2.c
output: Synchronized by the use of a shared variable to get value of Application ID*/

#include<stdio.h>
#include<omp.h>

int get_app_id(int l) /*function to assign Application ID to customer*/
{
return(l);

}
int get_main_token(int app_id) /*function to assign Main Counter Token ID to customer
according to his Application ID*/
{
return(app_id);
}
int get_sub_token(int x) /* function to assign Sub counter token ID to a customer after he
has been processed by the main counter*/
{
int k=x%5;
if(k==0)      /*when main token ID is a multiple of 5*/
return(k+5); /*customer will go to counter 5 as there is no counter 0 */
else
return(k);
}



main()
{
int app_id=0,main_token=0,sub_token=0,i,l=0;  /*int l is shared by default*/

#pragma omp parallel for firstprivate(app_id,main_token,sub_token)
for(i=1;i<=100;i++)
{
l=l+1;   /*shared int l is used to generate Application ID's */
app_id=get_app_id(l);  /*This causes the output to be in serial order irrespective of the
order of thread execution*/
main_token=get_main_token(app_id);
sub_token=get_sub_token(main_token);

printf("\nMain counter is currently processing Application ID:%d \nMain Counter Token ID :%d
\nCounter ID assigned:%d",app_id,main_token,sub_token);



}

}
```