In-Class Assignment: Peterson's Solution

Peterson's solution is a method that can be used to allow two processes to enter their critical section exclusively (only one process in their critical section at a time). For this in-class assignment you will create the pseudo code for two processes: "Process A" and "Process B". The two variables shared between the processes are **flag** and **turn**. Each process should attempt to enter its critical section and print "Hello World\n" using Peterson's solution.

Once you have the pseudocode written you will need to simulate the execution of each of the processes. Two simulations are expected, one in which process A will enter its critical section first and one in which process B will enter its critical section. Please use a different color font to represent each of the processes.

For example, if I have two processes that are context switched between them after the first process outputs "Hello World\n" with the printf statement.

Process A:

```
int main(){
        printf("Hello World\n");
        return 0;
}
```

Process B:

```
int main(){
        printf("Hello World\n");
        return 0;
}
```

Sample code execution:

```
int main(){
int main(){
printf("Hello World\n");
printf("Hello World\n");
return 0;
}
return 0;
}
```

*********Your work starts below here! **********

Process A pseudocode:

```
do {
        flag[a] = true;
        turn = b;
        while (flag[b] && turn = = b);
                critical section
        flag[a] = false;
                remainder section
 } while (true);
```

Process B pseudocode:

```
do {
        flag[b] = true;
        turn = a;
        while (flag[a] && turn = =a);
                critical section
        flag[b] = false;
                remainder section
} while (true);
```

Execution of commands where Process A enters the critical section first (needs to be completed):

```
do {
        flag[a] = true;
do {
        flag[b] = true;
```

Execution of commands where Process B enters the critical section first: