PRACTICAL NO. 6

AIM: Considered there are N philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pick up the two chopsticks adjacent to him. one chopstick may be picked up by any one of its adjacent followers but not both. Write a program to solve the problem using process synchronization technique.

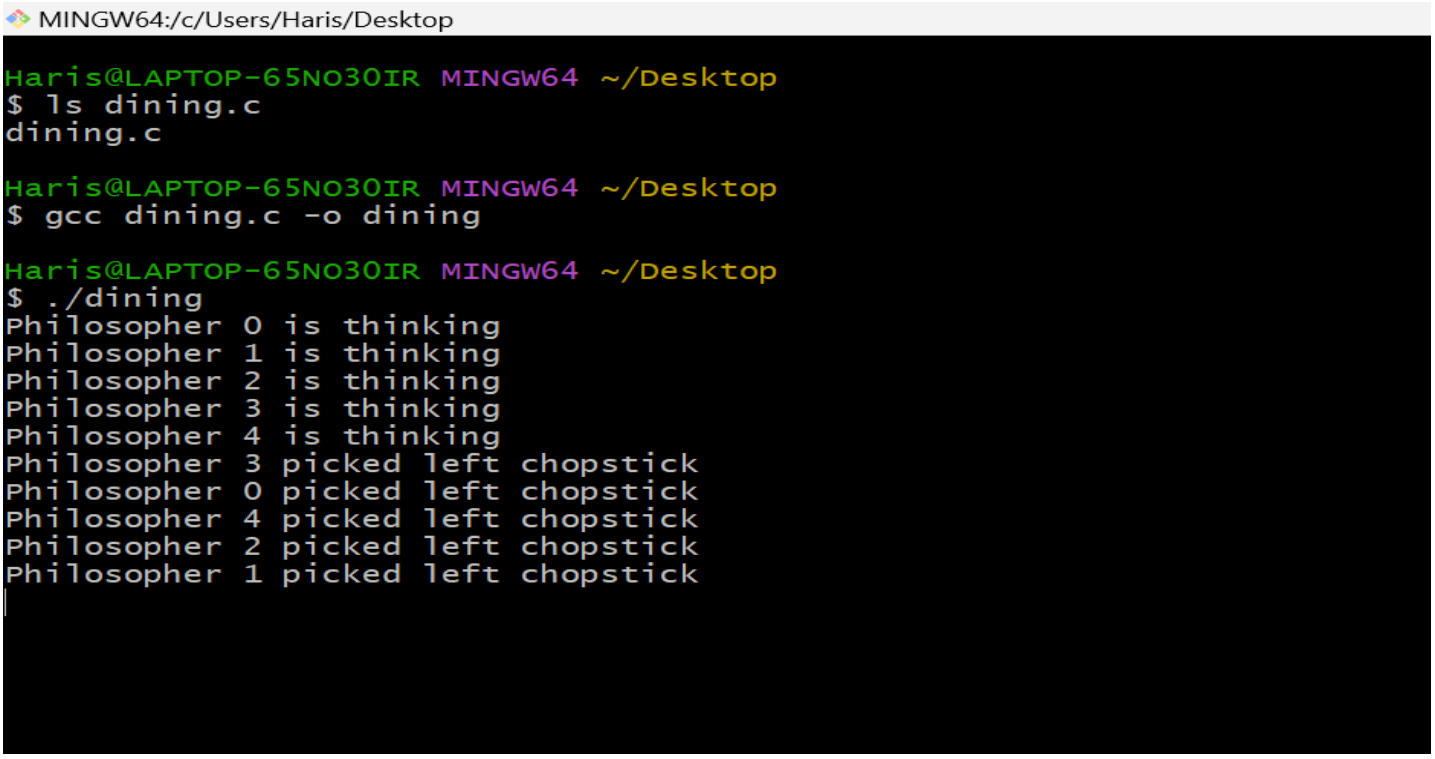
CODE:

MINGW64:/c/Users/Haris

GNU nano 8.7 dining.c

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

#define N 5

HANDLE chopstick[N];

DWORD WINAPI philosopher(LPVOID num) {
    int phil = *(int*)num;

    printf("Philosopher %d is thinking\n", phil);
    Sleep(1000);

    WaitForSingleObject(chopstick[phil], INFINITE);
    printf("Philosopher %d picked left chopstick\n", phil);

    WaitForSingleObject(chopstick[(phil+1)%N], INFINITE);
    printf("Philosopher %d picked right chopstick\n", phil);

    printf("Philosopher %d is eating\n", phil);
    Sleep(2000);

    ReleaseMutex(chopstick[phil]);
    ReleaseMutex(chopstick[(phil+1)%N]);

    printf("Philosopher %d finished eating\n", phil);

    return 0;
}

int main() {
    HANDLE thread[N];
    int phil_num[N];

    for(int i = 0; i < N; i++)
        chopstick[i] = CreateMutex(NULL, FALSE, NULL);

    for(int i = 0; i < N; i++) {
        phil_num[i] = i;
        thread[i] = CreateThread(NULL, 0, philosopher, &phil_num[i], 0, NULL);
    }

    WaitForMultipleObjects(N, thread, TRUE, INFINITE);

    return 0;
}
```

OUTPUT:


MINGW64:/c/Users/Haris/Desktop
Haris@LAPTOP-65NO3OIR MINGW64 ~/Desktop
$ ls dining.c
dining.c
Haris@LAPTOP-65NO3OIR MINGW64 ~/Desktop
$ gcc dining.c -o dining
Haris@LAPTOP-65NO3OIR MINGW64 ~/Desktop
$ ./dining
Philosopher 0 is thinking
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 3 picked left chopstick
Philosopher 0 picked left chopstick
Philosopher 4 picked left chopstick
Philosopher 2 picked left chopstick
Philosopher 1 picked left chopstick