DINING PHILOSOPHERS

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
#define N 5
sem_t forks[N];
void* philosopher(void* num){
   int i = *(int*)num;
   while(1){
      printf("Philosopher %d is Thinking\n", i+1);
      sem_wait(&forks[i]);
      sem_wait(&forks[(i+1)%N]);
      printf("Philosopher %d is Eating\n", i+1);
      sleep(1);
      sem_post(&forks[i]);
      sem_post(&forks[(i+1)%N]);
   }
}
int main(){
   pthread_t ph[N];
   int i, id[N];
   for(i=0;i<N;i++) sem_init(&forks[i],0,1);
   for(i=0;i<N;i++){ id[i]=i; pthread_create(&ph[i],NULL,philosopher,&id[i]); }
   for(i=0;i<N;i++) pthread_join(ph[i],NULL);
   return 0;
}
```