Assignment 4 Ryan Lebeau 104535367

1. - The code does guarantee mutual exclusion
     The code does guarantee progress
     The code is not designed for fairness but is very efficient
   - The code does not guarantee mutual exclusion
     The code does not guarantee progress
   - The code does guarantee mutual exclusion
     The code does not guarantee progress
   - The code does guarantee mutual exclusion
     The code does guarantee progress
     The code is designed for both fairness and efficiency, also it is designed as general as possible
2.
```
enter(){
        sem_t cust;
        sem_init(cust, 0, 1);
        sem_wait(&cust);
        //customer getting their number
        int customer = NumberGiven();
        sleep(1); //other calculations }
finish(){
        //customer number was called
        sem_post(&cust);
        sleep(1); //other calculations
        //ending semaphore
        sem_destroy(&cust); }
```
3. With signal() operations associated with monitors a thread ID is required to accomplish the operation. Where as with semaphores sem_wait() and sem_post(), the thread ID is not required, only the semaphore address.

4. pthread_mutex_t shiner;

```
int[N] chairs;
int cust;
shoeShiner(){
        pthread_mutex_lock(&shiner);
        sleep(3); //shining shoe
        pthread_mutex_unlock(&shiner);
        wait();//wait for more customers }
customer(){
        while(true){ //number of customers unknown
                error = pthread_create(&thread, NULL, &shoeShiner, NULL);
                if(error != 0 && cust < N){
                //someone has there shoe being shined and there are still seats
                        chairs.add(customer); }
                else{ continue;}
        cust++; }
```

5. pthread_mutex_t mut;

```
sem_t sem;
bool north = true;
char* lights;
main(){
        pthread_mutex_init(&mut, NULL);
        error = pthread_create(&thread, NULL, &arrive(north), NULL);
        if(error !=0) //lane is occupied
                printf("Wait");
        else{ //lane can change
                lights = "green";
                depart(north); }
}
```