Const int MAX\_CUST\_NO = 100;

Semaphore desk\_line = 10;

Semaphore desk = 1;

Semaphore desk\_cust\_ready = 0;

Semaphore agent\_line = 10;

Semaphore agent[2] = {1};

Semaphore agent\_cust\_ready = 0;

Semaphore announcer[MAX\_CUST\_NO] = {1};

Semaphore mutex1 = mutex2 = mutex3 = mutex4 = 1;

Semaphore exam [MAX\_CUST\_NO] = {1};

Semaphore finished [MAX\_CUST\_NO] = {0};

Int ticketnr[MAX\_CUST\_NO];

Int cust\_count, ticket\_count;

Void customer() {

Int custnr;

Int cust\_agent = (custnr % 2)

Wait(mutex1);

Cust\_count++

Custnr = cust\_count;

Singal(mutex1);

Wait(desk\_line);

Enter\_DMV();

Wait(desk);

Signal(desk\_line);

Wait(mutex2);

Enqueue1(custnr);

Signal(desk\_cust\_ready);

Signal(mutex2);

Enter\_waiting\_room();

Wait(announcer[ticketnr]);

Signal(waiting\_room);

Enter\_agent\_line();

Wait(agent[cust\_agent]);

Signal(agent\_line);

Wait(mutex4);

Enqueue2(custnr);

Signal(agent\_cust\_ready);

Signal(mutex4);

Signal(exam[custnr]);

Wait(finished[custnr]);

Signal(agent[cust\_agent]);

Exit();

}

Void information\_desk() {

Int desk\_cust;

While(true) {

Wait(desk\_cust\_ready);

Wait(mutex2);

Dequeue1(desk\_cust);

Signal(mutex2);

Wait(mutex3);

Ticket\_count++;

Ticketnr[desk\_cust] = ticket\_count;

Signal(mutex3);

Signal(desk);

}

}

Void announcer() {

While(true) {

Wait(agent\_line);

For(int x = 0; x < MAX\_CUST\_NO; x++) {

If(announcer[x] == 0) {

Signal(announcer[x]);

Break;

}

}

}

}

Void agent() {

Int agent\_cust;

While(true) {

Wait(agent\_cust\_ready);

Wait(mutex4);

Dequeue2(agent\_cust);

Signal(mutex4);

Wait(exam[agent\_cust]);

Signal(finished[agent\_cust]);

}

}