Pseudocode

**Semaphores:**

sem\_teller\_line,

sem\_officer\_line,

sem\_ready\_for\_teller,

sem\_ready\_for\_officer,

sem\_perform\_task,

sem\_employee\_assigned,

sem\_customer\_request\_teller,

sem\_customer\_request\_officer.

Customer {

while (visits != 3) {

enter\_bank(); //Create the Customer’s thread

get\_current\_task();

get\_current\_task\_amount();

wait(wait\_in\_line); //Add customer to queue

signal(wait\_in\_line);

signal(ready); //Customer is ready for employee

wait(assigned); //Customer waits for employee to be assigned

request\_task();

signal(request);

wait(perform\_task); //Customer is waiting while employee completed the task

get\_receipt();

visits++; //Go to next visit or leave the bank

}

}

Teller {

while(true) {

wait(ready\_for\_teller); //Waiting until customer will be ready

wait(wait\_in\_teller\_line); //Access to queue of customers

pull\_customer\_from\_line();

signal(wait\_in\_teller\_line); //Teller is finished using the queue

signal(assigned); //Signal to customer that employee was assigned

wait(request);

process\_task();

signal(perform\_task); //Signal to customer that task is done

}

}

LoanOfficer {

while(true) {

wait(ready\_for\_officer); //Waiting until customer will be ready

wait(wait\_in\_officer\_line); //Access to queue of customers

pull\_customer\_from\_line();

signal(wait\_in\_officer\_line); //Officer is finished using the queue

signal(assigned); //Signal to customer that employee was assigned

wait(request);

process\_loan();

signal(perform\_task); //Signal to customer that task is done

}

}

Main() {

create\_variables();

create\_semaphores();

create\_threads();//Create Customer’s, Teller’s and Officer’s threads

start\_threads();

join\_threads();

print\_summary();

}