// Devin Hardy

// Asg 4

// Barber shop problem with 2 threads, 2 mutexes, and a condition variable

#include <iostream>

#include <random>

#include <thread>

#include <mutex>

// Max number of chairs

int NumChairs = 4;

// Mutex for number of seats

//Conditions

// condition for number of customers in waiting

// count of number of seats

//barber algorithm:

/\*

def Barber():

loop

if customer is ready

awake

remove from seat 1 customer

unlock barber ready

unlock customer ready

cut hair

end

\*/

Barber{

while (true) {

//

}

}

//customer algorithm

/\*

def Customer():

loop

try to lock mutex protecting number of chairs

if number of seats > 0 then

decrement the number of available customer seats

signal the readyCustomer condition variable

Unlock the mutex protecting access the chairs

wait until the barber is ready

get hair cut

Else

unlock mutex protecting chairs

leave barbershop dissappointed

end

\*/

//Random number generator

// pass in first the min number and second the max number

// for the possible random number to be between

int RandNum(int first, int second)

{

std::random\_device dev;

std::mt19937 rng(dev());

std::uniform\_int\_distribution<std::mt19937::result\_type> dist6(first, second); // distribution in range [1, 6]

return (dist6(rng));

}

int main()

{

int runTime;

runTime = RandNum(100, 1000);

for (int i = 0; i < runTime; i++)

{

}

return 0;

}

/\*

Reference:

https://techtipqa.wordpress.com/2015/08/21/tech-qa-9-the-sleeping-barber-problem/

\*/