The assignment is intentionally vague about telling you exactly what output to produce for
the Sleeping Barber problem and there is no sample output.

It is important that the output from your programs be clear and able to convince everyone
that your program works correctly. In many programming situations, it seems that I spend
more time getting the output to print clearly than on the code itself. Once the output is
printing in a useful way, I can see exactly what the program is doing and debugging
proceeds quickly.

Here are some ideas for the output of the Sleeping Barber problem...

In the Producer/Consumer and Dining Philosophers problems, we tried to print the activity
of each thread in a different column so that you could see what each thread was doing.
Perhaps the actual events in this problem can be reduced to a single letter, such as:

        E = Enter
        S = Sit in waiting chair
        B = Begin haircut
        F = Finish haircut
        L = Leave
        start = Barber begins haircut
        end = Barber ends haircut

You might print the status of the chairs every time you print a line. If there are 5
chairs, of which 2 are occupied, print 2 X's and 3 -'s. For example:
        XX---

Here is some (hand-constructed) output to give you the idea. It prints best in fixed-
width font. In this example, I show only 10 customer threads.

```
        Barber   1  2  3  4  5  6  7  8  9  10
-----            E
X----            S
X----               E
XX---               S
X----    start
X----            B
X----                  E
XX---                  S
XX---                     E
XXX--                     S
XXX--                        E
XXXX-                        S
XXXX-                           E
XXXXX                           S
XXXXX                              E
XXXXX                              L
XXXXX            F
XXXXX    end
XXXX-    start
XXXX-                  B
XXXX-            L

etc...
```

As for the number of customer threads, you'll certainly want more than 5, since with 5 no one will ever be turned away.

I general, the more threads the better. With only 8, it might be likely that the first 3 people will be completed before the last 3 get there, meaning no one gets turned away.

I prefer to "hammer" on concurrent code. With this problem, I would probably create 20 customer threads and have each one of the customers iterate 10 times (i.e., get 10 haircuts in a row), for a total of 200 haircuts.

Note that each haircut may involve 7 lines of output:

        E
        S
        B
        start
        F
        end
        L

This would yield as many as 200*7 = a lot of lines of output. Can you think of a better approach?