Solution

1.
Using one lock and two semaphores.

```
Lock  lock;
Semaphore  barber;  // initialized to 1
Semaphore customers; //initialized to 0
int  empty_chairs = n;  //number of empty chairs
```

Pseudocode for customers:

```
    acquire(lock);
    if (empty_chairs == 0) {
        leave the barbershop;
        release(lock);
        return;
    }
    empty_chairs--;
    release(lock);
    signal(customers);
    wait(barber);
    have hairs being cut;
```

Pseudocode for barber:

```
while (1) {
    wait(customers);
    acquire(lock);
    take a customer;
    empty_chairs++;
    release(lock);
    cutting hair;
    signal(barber);
}
```

2.
This is about thrashing. Only d) and e) will likely improve CPU utilization.

3.
Assume access time to the associative memory is t nanoseconds.
Effective memory access time T = 0.8 * (1 microsecond + *t* nanosecond) +
0.18 * (2 microseconds + *t* nanosecond) +
0.02 * (2 microsecond + 20 milliseconds + *t* nanosecond)
= 401.2 microseconds (approximately)