For this assignment I was honestly a bit confused, especially with the first problem. What I ended up doing for problem one was having each guest (const N) go through the labyrinth and once every guest has gone through the labyrinth then each guest may eat their cupcake. I assigned a thread for the amount of guests attending. Testing out this problem to see if the number of guest goes up (this also means that the amount of threads also goes up) what happens to the run time. Run time consequently went up the more guest I added.

Problem two which I understood a little bit better I had the same structure of threads being equal to the number of guests. I also had an enumeration at the beginning of the program that consisted of two Occupancies, “Available” and “BUSY”. Essentially, I ended up using the second approach to solve this problem. The reason I went with this approach is because I felt like it was the most straightforward to implement. I felt like this approach would result to a faster runtime compared to the first approach since this first approach would lead to guest joining the room at the same time. Approach three could possibly be faster however guests would have to wait in line, limiting their freedom. The second approach as implemented would update the availability of the room in between the lock and unlock to ensure that only one thread is allowed in the room at a time. The runtime for this Problem was also similar to the first one.