Elijah Smith
COP4520 HW2
Problem 1: Birthday Party

The strategy that the guests should take to solve this problem is to designate one guest to take the role of keeping track of the total number of entered guests:

- Every guest, including this “special guest” will eventually have multiple chances to enter.
    - If they enter and find a cupcake waiting, they are to eat the cupcake and leave.
    - If any of the N-1 non-special guests enter and do NOT find a cupcake waiting, they should not request a new cupcake and should instead leave immediately.
    - This way, all N guests will eventually eat exactly one cupcake.
- The special guest additionally must:
    - If they enter the maze and find no cupcake waiting, make a note that a new unique guest has entered and eaten the cupcake. They should then call for the cupcake to be replaced.
- When the special guest has entered and noted that the cupcake was missing N-1 times, it is guaranteed that all guests have entered the maze at least once (and eaten the cupcake exactly once). The guests are now safe to tell the minotaur that all N guests have entered the maze.

Using the cupcake as a signal eliminates the need for guessing if any guest has or has not entered the maze. Each guest may enter the maze any number of times, but they eat the cupcake to signal exactly once that they have entered.

In order to implement this strategy, each guest is represented by a GuestThread and the special guest is represented by a SpecialGuestThread, which extends the GuestThread with the ability to count the total number of times the cupcake has been eaten. The main thread starts up all N threads at the same time and chooses randomly the next thread to be notified that it may now enter the Labyrinth, which performs the actions of that particular guest.

The following table shows the experimental results of the average number of times each member had to enter the Labyrinth before the safe termination was reached for varying values of N:

# Guests (N)	10	50	100	150	1000
Average Visits	12.5	52.24	100.2	136.18	997.544