Hungry, hungry hippos! There are $H$ hippos (4, in the board game, but we can imagine more). A referee dumps a handful of marbles ($M$ of them) between the hippos, and the hippos attempt to consume marbles as quickly as possible. However, each hippo can only consume one marble at a time. When all of the marbles have been consumed, the hippo who ate the last marble informs the referee, who prints the score for that round and then dumps $M$ more marbles onto the board. This game repeats forever.

The basic algorithm for the game is given below, but it lacks any synchronization. Fill in the appropriate synchronization and any conditionals (if-statements) required to protect the global data. You will need to declare your synchronization primitives, but you may assume they are correctly initialized elsewhere.

**Hint:** Use a monitor implemented with condition variables (CV wait and CV signal). Assume these condition variables follow the Mesa convention.

```c
hippo(int ID) {
    while (1) eat(ID);
}

referee() {
    while (1) restart_game();
}

// GLOBALS
int marbles = M;
int scores = {0, 0, 0, ... 0};

// SYNCHRONIZATION DECLARATIONS
```
void eat(int ID) {
    marbles--;
    scores[ID]++;
}

void restart_game() {
    print_and_clear_scores();
    marbles = M;
}