**Lab Assignment**

Design and implement a program to simulate a racing game among N players. The players move in sequence starting with player 1 and ending with player N, and the same cycle repeats as necessary. On each move, a player may move forward or backward depending on the player’s position change for the move. A positive position change implies a forward move, whereas a negative position change results in a backward move. As players move in sequence, the first player whose total move exceeds certain number of units (a given number, say, 50) wins, and the game is over. The position change a player receives on each move is determined by:

* The outcome of throwing a fair die (which can be simulated with a random number)
* The relative position of the player who makes a move with respect to the positions of other players.
* A move type the player currently uses; three types of move are defined as follows:

*Move Type 1:*

*change in position = die throw + ( position of leading player – player’s position ) / 2*

*If the die throw is 3, 4, 5, or 6, then the change in position is positive; otherwise it is negative.*

*Move Type 2:*

*change in position = (3 \* die throw) if die throw is even, or = (die throw) if die throw is odd.*

(Thus, the change in position is always positive.)

*Move Type 3*:

*change in position = die throw + (player’s position – position of the last player) / 2*

*If the die throw is 1 or 2, then the change in position is positive; otherwise it is negative.*

To start, a player picks a move type randomly (among type 1, type 2, and type 3 defined above), and another random number called *duration* (with a value between 2 and 5), and the player uses the move type picked for *duration* number of times. When this *duration* becomes zero, the player picks, at the next turn, another move type randomly with another random *duration*. As the contest progresses, the console output should display the player and its current position after each player’s move. When the game is over, the console output should display the winner of the game, and a table (of any format) that identifies the players and their final positions.

*Here is an example with 4 players:*

At some point, the four players are at positions shown in the following table:

Player 1 2 3 4

Position 10 5 12 7

Suppose player 3 is making a move. She currently uses move type 2 with *duration* = 1. So she throws a die (assuming it’s a 2). Thus her position change would be 3\*2 = 6 (according to the type 2 move logic). So her new position would be 18 (duration now becomes zero, so she would have to select another move type and duration at her next turn).

**Design consideration:** the design should accommodate program extension should new move types be added or old move types be removed later.