**<1> - <Incorrect payout>:**

Description: reproduces the bug that occurs when a player wins a bet with only one roll and their balance does not increase

Pre-conditions:

* balance > limit
* pick = only 1 roll

Post-conditions:

* balance after win is not more than balance before bet.

Data required:

* balance
* pick
* bet
* limit
* roll
* game
* winnings

**<2> - <player cannot reach betting limit>:**

Description: when the limit is set to 0, the games will still end when the player has 5 left. Reproduces this behavior.

Pre-conditions:

* balance > limit
* limit = 0
* balance = 5

Post-conditions:

* Game will end with the player still on 5 instead of playing to 0 if the bug is present

Data required:

* balance
* pick
* bet
* limit
* roll
* game
* winnings
* game state

**<3> - <odd in the game do not appear to be correct>:**

Description: the test case will run the program and trace the number of wins and losses to determine a ratio. If the bug is present the ratio will not be equal to 0.42. The test case will run a number of iterations of the program to work out if the ratio is consistent.

Pre-conditions:

* balance > limit

Post-conditions:

* Game will end and the ratio of wins and losses will be printed to the display.
* The win/loss ratio will not consistently be ~0.42 if the bug is present

Data required:

* Balance
* pick
* bet
* limit
* roll
* game
* winnings
* game state
* win/loss ratio

**<4> - <Game does not randomize winning rolls>:**

Description: The game appears to be selecting the same rolls every game. Randomizing them once at the start of the program and using the result for the remainder.

Pre-conditions:

* game at start.
* Rolls selected.

Post-conditions:

* all winning rolls will be the same if bug is present.

Data required:

* balance
* pick
* bet
* limit
* roll
* game
* winnings
* game state