CSCI1120 Test Cases
Assignment 3 Strings and Coins

1. Test cases

1.1 Function Unit Test for stringState

Test the stringState function with different conditions to verify the correctness of return value.

Case	Inputs	Outputs	Description
1	network=1111111111111111100, pos=1	true	Test network initial state as basic case
2	network=11111111111111100, pos=2	false	Test network value with leading zeros, test for cut string
3	network=11111111111111100, pos=3	true	Test network value with leading zeros, test for uncut string
4	network=11001000111111110, pos=4	true	Test network value with intermediate zeros, test for uncut string
5	network=11001000111111110, pos=5	false	Test network value with intermediate zeros, test for cut string
6	network=24, pos=17	false	Test network ultimate state as special case

1.2 Function Unit Test for playerScore

Test the playerScore function with different inputs to verify the functionality.

Case	Inputs	Outputs	Description
1	network=1000000031, p=1	3	Test whether this function can return the score of Player 1
2	network = 1000000031, p=2	1	Test whether this function can return the score of Player 2
3	network = 6, p = 2	6	A special case
4	network = 6, p = 1	0	A special case

1.3 Function Unit Test for updateNetwork

Test the updateNetwork function.

Case	Inputs	Outputs (network value)	Description
1	network=1111111111111111100, pos=1, p=1	111111111111111100	Test for player 1's first turn with network as initial state that leads to leading zero (basic case)
2	network=1111111111111111100, pos=3, p=1	1101111111111111100	Test for player 1's first turn with network as initial state that leads to intermediate zero (basic case)
3	network=11111000111111100, pos=5, p=2	11011000111111100	Test for player 2's intermediate turn with previous cut strings (basic case)
4	network=11001000111111010, pos=16, p=2	11001000111110010	Test for player 2's intermediate turn with existing disconnected coin (basic case)
5	network=111100012, pos=12, p=1	101100012	Test for player 1's intermediate turn with existing disconnected coin (basic case)
6	network=1010000111111101100, pos=9, p=1	1010000101111101110	Test for the case that player 1 score 1 point without previous disconnected coins (medium case)
7	network=10000101111101110, pos=8, p=2	10000001111101111	Test for the case that player 2 score 1 point with previous disconnected coins (medium case)
8	network=1011011111111100, pos=4, p=2	11011111111101	Test for the case that player 2 score 1 point without previous disconnected coins (medium case)
9	network=1110010002, pos=10, p=2	110010004	Test for the case that player 2 score 2 points in the intermediate turn (hard case)
10	network= 1000000031, pos=10, p=1	51	Test for the case that player 1 score 2 points at the last turn (hard case)

2.4 Gameflow Test

Here we use 2 test flows to test the implementation of gameflow.

2.4.1 Flow1

In flow1, player 1 wins. In this flow, we want to check whether this program can discover invalid inputs, print messages if player 1/2 disconnects coins, swap players, determine the final winner and print “Player 1 wins!”.

Step	Inputs	Output	Description
1	0 (player 1)	Invalid. Try again! Player 1’s move	Check the invalid input 0
2	18 (player 1)	Invalid. Try again! Player 1’s move	Check the invalid input 18
3	1 (player 1)	Player 2’s move	Cut string at pos 1. Swap the player
4	4 (player 2)	Player 1’s move	Cut string at pos 4. Swap the player
5	2 (player 1)	Player 2’s move	Cut string at pos 2. Swap the player
6	3 (player 2)	Player 1’s move	Cut string at pos 3. Swap the player
7	5 (player 1)	Player 2’s move	Cut string at pos 5. Swap the player
8	8 (player 2)	Player 2 score: 1 Player 2’s move	Check whether game add 1 score to player 2 and do n’t swap the player
9	5 (player 2)	Invalid. Try again! Player 2’s move	Check the invalid input which has been cut
10	6 (player 2)	Player 1’s move	Cut string at pos 6. Swap the player
11	7 (player 1)	Player 2’s move	Cut string at pos 7. Swap the player
12	11 (player 2)	Player 1’s move	Cut string at pos 11. Swap the player
13	9 (player 1)	Player 1 score: 1 Player 1’s move	Check whether game add 1 score to player 1 and do n’t swap the player
14	12 (player 1)	Player 2’s move	Cut string at pos 12. Swap the player
15	15 (player 2)	Player 2 score: 2 Player 2’s move	Check whether game add 1 score to player 2 and do n’t swap the player
16	14 (player 2)	Player 1’s move	Cut string at pos 14. Swap the player
17	17 (player 1)	Player 2’s move	Cut string at pos 17. Swap the player
18	13 (player 2)	Player 1’s move	Cut string at pos 10. Swap the player
19	10 (player 1)	Player 1 score: 3 Player 1’s move	Check whether game add 2 scores to player 1 and do n’t swap the player
20	16 (player 1)	Player 1 score: 4 Player 1 wins!	Determine the winner and output message

2.4.2 Flow2

In flow2, the game reaches a draw. In this flow, the step 1-17 are the same with flow 1. We want to use this flow to test draw game situation.

Step	Inputs	Output	Description
18	10 (player 2)	Player 2 score: 3 Player 2’s move	Check whether game add 1 score to player 2 and do n’t swap the player

19	16 (player 2)	Player 1's move	Cut string at pos 16. Swap the player
20	13 (player 1)	Player 1 score: 3 Draw game!	Check whether game add 2 scores to player 1 and output draw game