Problem statement

Create famous 'Frog leap' puzzle game. Try completing the game before starting to get an idea about its working. Demonstration.

Rules

1. The left set of frogs can only move right, the right set of frogs can only move left.
2. Frogs can move forward one space, or move two spaces by jumping over another frog from opposite side.
3. The puzzle is solved when the two sets of frogs have switched positions.

Steps to solve the problem:

Step1:-

- Display green and brown frogs on the left and right sides initially.

Initial Display :-

```
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['G', 'G', 'G', '-', 'B', 'B', 'B']
```

Here 'G' represents Green frogs on the left side and 'B' represents brown frogs on the right side. The '-' defines the position of empty leaf. (You can change display according to your imagination or convinience)

Step2:-

Accept positions of the frog that you want to move.
Example: If we enter position 2 then the game will look like this:-

```
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['G', 'G', '-', 'G', 'B', 'B', 'B']
```

Step3:-

Define Invalid moves and add conditional 'if' statements accordingly

Rules

1. Entered position should be between 0 to 6. Or a character 'q' to quit the game.
2. Entered position cannot be the position of empty leaf.
3. If the selected frog position cannot perform the contraints given in rule 2 then the move is invalid.

Step4:-

Make the appropriate move by changing the game display.

Step 1

First create a list `positions` which contains the characters 'G','B' and '-' in the same sequence as given in the initial display state.

```
### your code here
positions = list(['🟢','🟢','🟢','--','🟤','🟤','🟤'])
```

Now print this string `[ 0 , 1 , 2 , 3 , 4 , 5 , 6 ]` and after that print the list `positions`

```
### your code here
print('[ 0 , 1 , 2 , 3 , 4 , 5 , 6 ]')
print(positions)
```

```
[ 0 , 1 , 2 , 3 , 4 , 5 , 6 ]
['🟢', '🟢', '🟢', '--', '🟤', '🟤', '🟤']
```

Take position input from user and write a message as `"Press q to quit else \nEnter position of piece:"`.

```
### your code here
pos = input("Press q to Quit \nEnter position of piece to move:")
```

```
Press q to Quit
Enter position of piece to move:2
```

Now the taken input is in string format. So first check if the input is `'q'` character. If input is `'q'` then the person is quiting the game so print `'You Lose'`.

```
### your code here
if pos == 'q':
  print('You Lose :(')
```

Next if input character is not `'q'` then it has to be some integer. so convert input to integer format.

```
### your code here
pos = int(pos)
```

Step 2

Now we have to check validity of the selected positions or move.
If the entered number isn't between 0 and 6, then print 'Invalid move'.

```
### your code here
if pos < 0 or pos > 6:
  print('Invalid Move!')
```

A frog should be present on the selected position to make a move. If leaf is selected then it doesn't make sense. Therefore, if entered postition is same as the postition of empty leaf then the move is invalid and print `Invalid Move`

```
### your code here
if positions[pos] == '--':
  print('Invalid Move!')
```

Initialize a variable named `pos2` at value 0, to store the index of empty leaf, so that we can use it later.

```
### your code here
pos2 = 0
```

```
Check if the selected frog is 'G':

    (Inside if when it's 'G'. As 'G' is selected frog can move to right only.)

     ❗condition 1

    If **selected_position + 1** is less than or equal to 6 and **curent_position + 1** contains '-'
    then it's a valid move and store that postion in `pos2`.

     ❗condition2

    Else if **selected_position + 2** is less than or equal to 6 and if **current_position + 2**
    contains '-' and if **selected_position + 1** contains 'B' then it's a valid move  and store that postion in `pos2`.

     ❗condition3:

    Else remainig all are invalid, so print `Invalid Move`
```

```
### your code here
if positions[pos] == '🟢':
  if (pos + 1 ) <= 6 and positions[pos + 1] == '--':
    pass
  elif(pos + 2) <=6 and positions[pos + 2] == '--' and positions[pos + 1] == '🟤':
    pass
  else:
    print("Invalid Move")
```

Check if the selected frog is 'B':

(Inside if when it's 'B'. As 'B' is selected frog can move to left only.)

❗ condition1:

If **selected_position - 1** is more than or equal to 0 and **curent_position - 1** contains '-' then it's a valid move and and store that postion in `pos2`.

❗ condition2:

Else if **selected_position - 2** is more than or equal to 0 and if **current_position - 2** contains '-' and if **selected_position - 1** contains 'G' then it's a valid move and and store that postion in `pos2`.

❗ condition3:

Else remainig all are invalid,, so print `Invalid Move`.

```
### your code here
if positions[pos] == '🟤':
  if (pos - 1) >= 0 and positions[pos - 1] == '--':
    pass
  elif (pos - 2) >= 0 and positions[pos - 2] == '--' and positions[pos - 1] == '🟢':
    pass
  else:
    print('Invalid Move')
```

Swap the element at selected positions and calculated position2 in the list.
So basically we are moving the frog to next valid position by swapping elelments of array.

```
### your code here
positions[pos], positions[pos2] = positions[pos2], positions[pos]
```

Now print the display of the game again to see the change.
If we enter position 2 then the output will look like this:-

```
 [ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
 ['G', 'G', '-', 'G', 'B', 'B', 'B']
```

```
### your code here
print('[ 0 , 1 , 2 , 3 , 4 , 5 , 6 ]')
print(positions)
```

```
    [ 0 , 1 , 2 , 3 , 4 , 5 , 6 ]
    ['🟢', '🟢', '🟢', '--', '🟤', '🟤', '🟤']
```

Check for winning condition by comparing the elements of list. If player has won the game print `'You Win'`

```
### your code here
if positions == ['🟢','🟢','🟢','--','🟤','🟤','🟤']:
  print(' ')
  print('You Win!')
```

```
    You Win!
```

Now the game should keep running until the player quits, so place all conditional statements inside an infinite loop.

1. We have to `'break'` the loop if the player presses `'q'` and quits.
2. If the move made by player is `'Invalid Move'` then we have to `'continue'` without executing remaining part of the selected iteration.
3. If player wins the game we have to `break` the loop.

```
 Infinite loop:
    (inside loop)
    1.Take input
    2.Check all valid and invalid conditions of `pos`.
```

3.Make the appropriate move by calculating `pos2`.

4.Display game

4.Check winning condition

```
positions = list(['🟢','🟢','🟢','--','🟤','🟤','🟤'])
print("[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]")
print(positions)
print(" ")

while True:
    pos = input("Press q to Quit \nEnter position of piece to move: ")
    if pos == 'q':
        print("You Lose!")
        print(" ")
        break
    pos = int(pos)

    if pos < 0 or pos > 6:
        print("Invalid Move")
        print(" ")
        continue
    if positions[pos] == '--':
        print("Invalid Move")
        print(" ")
        continue

    if positions[pos] == '🟢':
        if (pos + 1) <= 6 and positions[pos + 1] == '--':
            pass
        elif (pos + 2) <= 6 and positions[pos + 2] == '--' and positions[pos + 1] == '🟤':
            pass
        else:
            print("Invalid Move")
            break
    if positions[pos] == '🟤':
        if (pos - 1) >= 0 and positions[pos - 1] == '--':
            pass
        elif (pos - 2) >= 0 and positions[pos - 2] == '--' and positions[pos - 1] == '🟢':
            pass
        else:
            print("Invalid Move")
            break

    pos2 = 0
    if positions[pos] == '🟢':
        if positions[pos + 1] == '--':
            pos2 = (pos+1)
        elif positions[pos + 2] == '--':
            pos2 = (pos+2)
    if positions[pos] == '🟤':
        if positions[pos - 1] == '--':
            pos2 = (pos-1)
        elif positions[pos - 2] == '--':
            pos2 = (pos-2)
    positions[pos], positions[pos2] = positions[pos2], positions[pos]

    print("[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]")
    print(positions)
    print(" ")

    if positions == ['🟤','🟤','🟤','--','🟢','🟢','🟢']:
        print(" ")
        print("You Win!")
        break
```

```
Press q to Quit
Enter position of piece to move: 4
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟢', '🟤', '🟢', '--', '🟢', '🟤']

Press q to Quit
Enter position of piece to move: 6
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟢', '🟤', '🟢', '🟤', '🟢', '--']

Press q to Quit
Enter position of piece to move: 5
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟢', '🟤', '🟢', '🟤', '--', '🟢']

Press q to Quit
Enter position of piece to move: 3
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟢', '🟤', '--', '🟤', '🟢', '🟢']

Press q to Quit
Enter position of piece to move: 1
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '--', '🟤', '🟢', '🟤', '🟢', '🟢']

Press q to Quit
Enter position of piece to move: 2
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟤', '--', '🟢', '🟤', '🟢', '🟢']

Press q to Quit
Enter position of piece to move: 4
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟤', '🟤', '🟢', '--', '🟢', '🟢']

Press q to Quit
Enter position of piece to move: 3
[ 0 ,  1 ,  2 ,  3 ,  4 ,  5 ,  6 ]
['🟤', '🟤', '🟤', '--', '🟢', '🟢', '🟢']


You Win!
```