IT314 Software Engineering

LAB 7

Program Inspection, Debugging and Static Analysis

-ID: 202201071
-Name: Nirva Patel

1 Armstrong

A. Program Inspection

1. There is one error in the program, related to the computation of the remainder, and it has been identified and corrected.
2. The most effective category of program inspection for this code is Category C: Computation Errors, as the error pertains to the computation of the remainder, a type of computation error.
3. Program inspection does not identify debugging-related errors. It does not detect issues such as breakpoints or runtime errors like logic errors.
4. The program inspection technique is valuable for identifying and rectifying issues related to code structure and computation errors.

B. Debugging

1. There is one error in the program related to the computation of the remainder, as previously identified.
2. To fix this error, one should set a breakpoint at the point where the remainder is computed to ensure it's calculated correctly. Step through the code to observe the values of variables and expressions during execution.
3. The corrected executable code is as follows:

```
// Armstrong Number class Armstrong {
    public static void main(String args[]) { int  num  =  Integer.parseInt(args[0]);
        int n = num; // used to check at the last time int check = 0, remainder;
        while (num > 0) { remainder = num % 10;
            check = check + (int) Math.pow(remainder, 3); num = num / 10;
        }
        if (check == n)
            System.out.println(n + " is an Armstrong Number"); else
            System.out.println(n + " is not an Armstrong Number");
    }
}
```

2 GCD and LCM

A. Program Inspection

1. There are two errors in the program:
2. Error 1: In the gcd function, the while loop condition should be while(a % b != 0) instead of
   while(a % b == 0) to calculate the GCD correctly.

3. Error 2: In the lcm function, there is a logic error. The logic used to calculate LCM is incorrect and will result in an infinite loop.
4. For this code, the most effective category of program inspection is Category C: Computation Errors, as it contains computation errors in both the gcd and lcm functions.
5. Program inspection is not able to identify runtime issues or logical errors. It can't identify errors like infinite loops.
6. The program inspection technique is worth applying to identify and fix computation-related issues.

B. Debugging

1. There are two errors in the program as mentioned above.
2. To fix these errors:
3. For Error 1 in the gcd function, you need one breakpoint at the beginning of the while loop to verify the correct execution of the loop.
4. For Error 2 in the lcm function, you would need to review the logic for calculating LCM, as it's a logical error.
5. The corrected executable code is as follows:

```
import java.util.Scanner;

public class GCD_LCM {
	static int gcd(int x, int y) { int a, b;
		a = (x > y) ? x : y; // a is greater number
		b = (x < y) ? x : y; // b is smaller number
		while (b != 0) { // Fixed the while loop condition int temp = b;
			b = a % b; a = temp;
		}
		return a;
	}

	static int lcm(int x, int y) {
		return (x * y) / gcd(x, y); // Calculate LCM using GCD
	}

	public static void main(String args[]) { Scanner input = new Scanner(System.in);
		System.out.println("Enter the two numbers: "); int x = input.nextInt();
		int y = input.nextInt();

		System.out.println("The GCD of two numbers is: " + gcd(x, y));
		System.out.println("The LCM of two numbers is: " + lcm(x, y)); input.close();
	}
}
```

3 Knapsack

A. Program Inspection

1. There is one error in the program. It is in the following line: int option1 = opt[n++][w]; The variable n is incremented, which is not intended. It should be: int option1 = opt[n][w];

2. The category of program inspection that would be most effective for this code is Category C: Computation Errors, as the identified error is related to computation within loops.

3. Program inspection is not able to identify runtime errors or logical errors that might arise during program execution.

4. The program inspection technique is worth applying to identify and fix computation-related issues.

B. Debugging

1. There is one error in the program, as identified above.

2. To fix this error, you would need one breakpoint at the line: int option1 = opt[n][w]; to ensure n and w are correctly used without unintended increments.

3. The corrected executable code is as follows:

```
public class Knapsack {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);          // number of items
        int W = Integer.parseInt(args[1]);          // maximum weight of knapsack

        int[] profit = new int[N + 1]; int[] weight = new int[N + 1];

        // Generate random instance, items 1..N for (int n = 1; n <= N; n++) {
            profit[n] = (int) (Math.random() * 1000); weight[n] = (int) (Math.random() * W);
        }

        int[][] opt = new int[N + 1][W + 1]; boolean[][] sol = new boolean[N + 1][W + 1];

        for (int n = 1; n <= N; n++) {
            for (int w = 1; w <= W; w++) {
                int option1 = opt[n - 1][w]; // Fixed the increment here int option2 =
                Integer.MIN_VALUE;
                if (weight[n] <= w)
                                option2 = profit[n] + opt[n - 1][w - weight[n]];

                opt[n][w] = Math.max(option1, option2); sol[n][w] = (option2 > option1);
            }
        }
```

```
        // Rest of the code is fine

        // Print results
        System.out.println("Item" + "\t" + "Profit" + "\t" + "Weight" + "\t" + "Take"); for (int n
        = 1; n <= N; n++) {
            System.out.println(n + "\t" + profit[n] + "\t" + weight[n] + "\t" + take[n]);
        }
    }
}
```

4 Magic Number

A. Program Inspection

1. There are two errors in the program:
2. Error 1: In the inner while loop, the condition should be while (sum > 0) instead of while (sum == 0).
3. Error 2: Inside the inner while loop, there are missing semicolons in the lines: s=s*(sum/10); sum=sum%10
   They should be corrected as: s = s * (sum / 10); sum = sum % 10;
4. The category of program inspection that would be most effective for this code is Category C: Computation Errors, as it contains computation errors in the while loop.
5. Program inspection is not able to identify runtime issues or logical errors that might arise during program execution.
6. The program inspection technique is worth applying to identify and fix computation-related issues.

B. Debugging

1. There are two errors in the program, as identified above.
2. To fix these errors, you would need one breakpoint at the beginning of the inner while loop to verify the execution of the loop. You can also use breakpoints to check the values of num and s during execution.
3. The corrected executable code is as follows:

```
import java.util.*;

public class MagicNumberCheck {
    public static void main(String args[]) { Scanner ob = new Scanner(System.in);
        System.out.println("Enter the number to be checked."); int n = ob.nextInt();
        int sum = 0, num = n; while (num > 9) {
            sum = num; int s = 0;
```

```
            while (sum > 0) { // Fixed the condition here s = s * (sum / 10);
                sum = sum % 10; // Fixed the missing semicolon
            }
            num = s;
        }
        if (num == 1) {
            System.out.println(n + " is a Magic Number.");
        } else {
            System.out.println(n + " is not a Magic Number.");
        }
    }
}
```

5 Merge Sort

A. Program Inspection

1. There are several errors in the program:
2. Error 1: In the mergeSort method, the lines int[] left = leftHalf(array+1); and int[] right = rightHalf(array-1); should be corrected. It seems like an attempt to split the array, but it's not done correctly.
3. Error 2: The leftHalf and rightHalf methods are incorrect. They should return the correct halves of the array.
4. Error 3: The merge method should have left and right arrays as inputs, not left++ and right--.
5. The category of program inspection that would be most effective for this code is Category C: Computation Errors, as there are computation-related issues in the code.
6. Program inspection cannot identify runtime issues or logical errors that might arise during program execution.
7. The program inspection technique is worth applying to identify and fix computation-related issues.

B. Debugging

1. There are multiple errors in the program, as identified above.
2. To fix these errors, you would need to set breakpoints to examine the values of left, right, and array during execution. You can also use breakpoints to check the values of i1 and i2 inside the merge method.

3. The corrected executable code is as follows:

```
import java.util.*; public class MergeSort {
    public static void main(String[] args) {
        int[] list = {14, 32, 67, 76, 23, 41, 58, 85};
```

```
        System.out.println("before: " + Arrays.toString(list)); mergeSort(list);
        System.out.println("after:          " + Arrays.toString(list));
    }

    public static void mergeSort(int[] array) { if (array.length > 1) {
            int[] left = leftHalf(array); int[] right = rightHalf(array); mergeSort(left);
            mergeSort(right);
            merge(array, left, right);
        }
    }

    public static int[] leftHalf(int[] array) { int size1 = array.length / 2;
        int[] left = new int[size1];
        for (int i = 0; i < size1; i++) { left[i] = array[i];
        }
        return left;
    }

    public static int[] rightHalf(int[] array) { int size1 = array.length / 2;
        int size2 = array.length - size1; int[] right = new int[size2];
        for (int i = 0; i < size2; i++) { right[i] = array[i + size1];
        }
        return right;
    }

    public static void merge(int[] result, int[] left, int[] right) { int i1 = 0;
        int i2 = 0;
        for (int i = 0; i < result.length; i++) {
            if (i2 >= right.length || (i1 < left.length && left[i1] <= right[i2])) { result[i] = left[i1];
                i1++;
            } else {
                result[i] = right[i2]; i2++;
            }
        }
    }
}
```

6 Multiply Matrices

A. Program Inspection

1. There are several errors in the program:

2. Error 1: In the nested loops for matrix multiplication, the loop indices should start from 0, not
   -1.

3. Error 2: The error message when the matrix dimensions are incompatible should print ”Matrices with entered orders can’t be multiplied with each other,” not ”Matrices with entered orders can’t be multiplied with each other.”

4. The category of program inspection that would be most effective for this code is Category C: Computation Errors, as there are computation-related issues in the code.

5. Program inspection cannot identify runtime issues or logical errors that might arise during program execution.

6. The program inspection technique is worth applying to identify and fix computation-related issues.

B. Debugging

1. There are multiple errors in the program, as identified above.

2. To fix these errors, you would need to set breakpoints to examine the values of c, d, k, and sum during execution. You should pay particular attention to the nested loops where the matrix multiplication occurs.

3. The corrected executable code is as follows:

```
import java.util.Scanner; class MatrixMultiplication {
    public static void main(String args[]) {
        int m, n, p, q, sum = 0, c, d, k;

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of the first matrix");
        m  =  in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];
        System.out.println("Enter the elements of the first matrix"); for (c = 0; c < m; c++)
            for (d = 0; d < n; d++) first[c][d] = in.nextInt();

        System.out.println("Enter the number of rows and columns of the second matrix");
        p = in.nextInt();
        q = in.nextInt();
```

```
        if (n != p)
            System.out.println("Matrices with entered orders can't be multiplied
            with each other."); else {
            int second[][] = new int[p][q];
            int multiply[][] = new int[m][q];
            System.out.println("Enter the elements of the second matrix"); for (c = 0; c <
            p; c++)
                for (d = 0; d < q; d++) second[c][d] = in.nextInt();

                                        for (c = 0; c < m; c++) {
                                    for (d = 0; d < q; d++) {
                                        for (k = 0; k < p; k++) {
                                        sum = sum + first[c][k] * second[k][d];
                        }

                        multiply[c][d] = sum; sum = 0;
                    }
            }

            System.out.println("Product of entered matrices:-");

            for (c = 0; c < m; c++) { for (d = 0; d < q; d++)
                        System.out.print(multiply[c][d]  +  "\t");

                    System.out.print("\n");
            }
        }
    }
}
```

7 Quadratic Probing

A. Program Inspection

1. There are multiple errors in the program:
2. Error 1: The insert method has a typo in the line i + = (i + h / h–)
3. Error 2: In the remove method, there is a logic error in the loop to rehash keys. It should be i = (i + h * h++)
4. Error 3: In the get method, there is a logic error in the loop to find the key. It should be i = (i
   + h * h++)
5. The category of program inspection that would be most effective for this code is Category A: Syntax Errors and Category B: Semantic Errors, as there are both syntax errors and semantic issues in the code.

6. The program inspection technique is worth applying to identify and fix these errors, but it may not identify logical errors that affect the program's behavior.

B. Debugging

1. There are three errors in the program, as identified above.

2. To fix these errors, you would need to set breakpoints and step through the code while examining variables like i, h, tmp1, and tmp2. You should pay attention to the logic of the insert, remove, and get methods.

3. The corrected executable code is as follows:

```
import java.util.Scanner;

class QuadraticProbingHashTable { private int currentSize, maxSize; private String[] keys;
    private String[] vals;

    public QuadraticProbingHashTable(int capacity) { currentSize = 0;
        maxSize = capacity;
        keys = new String[maxSize]; vals = new String[maxSize];
    }

    public void makeEmpty() { currentSize = 0;
        keys = new String[maxSize]; vals = a String[maxSize];
    }

    public int getSize() { return currentSize;
    }

    public boolean isFull() {
        return currentSize  ==  maxSize;
    }

    public boolean isEmpty() { return getSize() == 0;
    }

    public boolean contains(String key) { return get(key) != null;
    }

    private int hash(String key) { return key.hashCode() % maxSize;
    }
```

```
public void insert(String key, String val) { int tmp = hash(key);
	int i = tmp, h = 1; do {
		if (keys[i] == null) { keys[i] = key; vals[i] = val; currentSize++; return;
		}
		if (keys[i].equals(key)) { vals[i] = val;
			return;
		}
		i += (h * h++) % maxSize;
	} while (i != tmp);
}

public String get(String key) { int i = hash(key), h = 1; while (keys[i] != null) {
		if (keys[i].equals(key)) return vals[i];
		i = (i + h * h++) % maxSize;
	}
	return null;
}

public void remove(String key) { if (!contains(key))
		return;

	int i = hash(key), h = 1; while (!key.equals(keys[i]))
		i = (i + h * h++) % maxSize; keys[i] = vals[i] = null;

	for (i = (i + h * h++) % maxSize; keys[i] != null; i = (i + h * h++) % maxSize)
	{
		String tmp1 = keys[i], tmp2 = vals[i]; keys[i] = vals[i] = null;
		currentSize--; insert(tmp1, tmp2);
	}
	currentSize--;
}

public void printHashTable() { System.out.println("\nHash Table: "); for (int i = 0; i <
	maxSize; i++)
		if (keys[i] != null)
```

```
                System.out.println(keys[i] + " " + vals[i]); System.out.println();
        }
}

public class QuadraticProbingHashTableTest { public static void main(String[] args) {
        Scanner scan = new Scanner(System.in); System.out.println("Hash Table
        Test\n\n"); System.println("Enter  size");

        QuadraticProbingHashTable qpht = new

        QuadraticProbingHashTable(scan.nextInt()); char ch;

        do {
            System.out.println("\nHash Table Operations\n"); System.out.println("1.
            insert"); System.out.println("2. remove"); System.out.println("3. get");
            System.out.println("4. clear"); System.out.println("5. size");

            int choice = scan.nextInt();

            switch (choice) { case 1:
                    System.out.println("Enter key and value"); qpht.insert(scan.next(),
                    scan.next()); break;
                case 2:
                    System.out.println("Enter key"); qpht.remove(scan.next());
                    break; case 3:
                    System.out.println("Enter key"); System.out.println("Value = " +
                    qpht.get(scan.next())); break;
                case 4:
                    qpht.makeEmpty();
                    System.out.println("Hash Table Cleared\n"); break;
                case 5:
                    System.out.println("Size = " + qpht.getSize()); break;
                default:
                    System.out.println("Wrong Entry\n"); break;
            }
            qpht.printHashTable();
            System.out.println("\nDo you want to continue (Type y or n) \n"); ch =
            scan.next().charAt(0);
```

```
        } while (ch == 'Y' || ch == 'y');
    }
}
```

8 Sorting Array

A. Program Inspection

1. Errors identified:

2. Error 1: The class name ”Ascending Order” contains an extra space and an underscore. The class name should be corrected to ”AscendingOrder.”

3. Error 2: The first nested for loop has an incorrect loop condition for (int i = 0; i ¿= n; i++);, which should be modified to for (int i = 0; i ¡ n; i++).

4. Error 3: There is an extra semicolon (;) after the first nested for loop, which should be removed.

5. The most effective category of program inspection would be Category A: Syntax Errors and Category B: Semantic Errors, as there are both syntax errors and semantic issues in the code.

6. Program inspection alone can identify and fix syntax errors and some semantic issues. However, it may not detect logic errors that affect the program’s behavior.

7. The program inspection technique is worth applying to fix the syntax and semantic errors, but debugging is required to address logic errors.

B. Debugging

1. There are two errors in the program as identified above.

2. To fix these errors, you need to set breakpoints and step through the code. You should focus on the class name, the loop conditions, and the unnecessary semicolon.

3. The corrected executable code is as follows:

```
import java.util.Scanner; public class AscendingOrder {
    public static void main(String[] args) {
        int n, temp;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of elements you want in the array: "); n =
        s.nextInt();
        int a[] = new int[n]; System.out.println("Enter all the elements:"); for (int i = 0; i < n;
        i++) {
            a[i] = s.nextInt();
        }
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) { if (a[i] > a[j]) {
                    temp = a[i]; a[i] = a[j];
```

```
                        a[j] = temp;
                    }
                }
            }
            System.out.print("Ascending Order: "); for (int i = 0; i < n - 1; i++) {
                System.out.print(a[i] + ", ");
            }
            System.out.print(a[n - 1]);
        }
}
```

9 Stack Implementation

A. Program Inspection

1. Errors identified:

2. Error 1: The push method has a decrement operation on the top variable (top–) instead of an increment operation. It should be corrected to top++ to push values correctly.

3. Error 2: The display method has an incorrect loop condition in for(int i=0; i ¿ top; i++). The loop condition should be for (int i = 0; i ¡= top; i++) to correctly display the elements.

4. Error 3: The pop method is missing in the StackMethods class. It should be added to provide a complete stack implementation.

5. The most effective category of program inspection would be Category A: Syntax Errors, as there are syntax errors in the code. In addition, Category B: Semantic Errors can help identify logic and functionality issues.

6. The program inspection technique is worth applying to identify and fix syntax errors, but additional inspection is needed to ensure the logic and functionality are correct.

B. Debugging

1. There are three errors in the program, as identified above.

2. To fix these errors, you would need to set breakpoints and step through the code, focusing on the push, pop, and display methods. Correct the push and display methods and add the missing pop method to provide a complete stack implementation.

3. The corrected executable code is as follows:

```
public class StackMethods { private int top;
    int size; int[] stack;

    public StackMethods(int arraySize) { size = arraySize;
        stack = new int[size]; top = -1;
    }
```

```
    public void push(int value) { if (top == size - 1) {
            System.out.println("Stack is full, can't push a value");
        } else {
            top++;
            stack[top] = value;
        }
    }

    public void pop() { if (!isEmpty()) {
            top--;
        } else {
            System.out.println("Can't pop...stack is empty");
        }
    }

    public boolean isEmpty() { return top == -1;
    }

    public void display() {
        for (int i = 0; i <= top; i++) { System.out.print(stack[i] + " ");
        }
        System.out.println();
    }
}
```

10 Tower of Hanoi

A. Program Inspection

1. Errors identified:
2. Error 1: In the line doTowers(topN ++, inter–, from+1, to+1), there are errors in the increment and decrement operators. It should be corrected to doTowers(topN - 1, inter, from, to).
3. The most effective category of program inspection would be Category B: Semantic Errors because the errors in the code are related to logic and function.
4. The program inspection technique is worth applying to identify and fix semantic errors in the code.

B. Debugging

1. There is one error in the program, as identified above.
2. To fix this error, you need to replace the line:

doTowers(topN ++, inter--, from+1, to+1);

3. with the correct version:

```
doTowers(topN - 1, inter, from, to);
```

4. The corrected executable code is as follows:

```
public class MainClass {
    public static void main(String[] args) { int nDisks = 3;
        doTowers(nDisks, 'A', 'B', 'C');
    }

    public static void doTowers(int topN, char from, char inter, char to) { if (topN == 1) {
            System.out.println("Disk 1 from " + from + " to " + to);
        } else {
            doTowers(topN - 1, from, to, inter);
            System.out.println("Disk " + topN + " from " + from + " to " + to); doTowers(topN -
            1, inter, from, to);
        }
    }
}
```

Acknowledgement: Was unable to find 2000 lines of code, so answered the questions using small fragments of code that summed up to approximately 2000.

CODE 1 :
Github link:
https://github.com/Medium/medium-sdk-nodejs/blob/master/test/mediumClient_test.js

```
// Import necessary modules

var medium = require("../") // Import the Medium client module
var nock = require("nock") // Mock HTTP requests
var qs = require('querystring') // Query string module for parsing and
formatting URLs
var should = require("should") // Assertion library for testing
var url = require('url') // URL module for handling URLs

var clientId = 'xyz' // Client ID for Medium API

// Test suite for MediumClient constructor
describe('MediumClient - constructor', function () {

  // Test: Should throw an error when options are undefined
  it('should throw a MediumError when options are undefined', function (done)
{
    (function () {
      if (undefinedVar) {
        // Attempt to instantiate without options, which should trigger an
error
        new medium.MediumClient()
      }
    }).should.throw(medium.MediumError) // Check if the correct error is
thrown
    done()
  })

  // Test: Should throw an error when options are empty
  it('should throw a MediumError when options are empty', function (done) {
    (function () {
      var options = {};
      if (options === "") {
        new medium.MediumClient(options) // This should trigger an error due
to missing required fields
      }
    }).should.throw(medium.MediumError) // Ensure the right error is thrown
    done()
  })

  // Test: Should throw an error if only clientId is provided
  it('should throw a MediumError when only clientId is provided', function
(done) {
```

```
    (function () {
      // Instantiating without clientSecret should throw an error
      new medium.MediumClient({ clientId: 'xxx' })
    }).should.throw(medium.MediumError)
    done()
  })

  // Test: Should throw an error if only clientSecret is provided
  it('should throw a MediumError when only clientSecret is provided', function
(done) {
    (function () {
      // Instantiating without clientId should trigger an error
      new medium.MediumClient({ clientSecret: 'yyy' })
    }).should.throw(medium.MediumError)
    done()
  })

  // Test: Should succeed when both clientId and clientSecret are provided
  it('should succeed when both clientId and clientSecret are provided',
function (done) {
    var client = new medium.MediumClient({ clientId: 'xxx', clientSecret:
'yyy' }) // Valid instantiation
    done() // Mark test as done
    // Control-Flow Error: Code after done() call can lead to undefined
behavior in test execution
    console.log("This should not run after done()") // This line should not
execute after done is called
  })
})

// Test suite for MediumClient methods
describe('MediumClient - methods', function () {

  // Redeclaring clientId here causes a data declaration error
  var clientId = 'xxx' // Data-Declaration Error
  var clientSecret = 'yyy'
  var client

  // Setup before each test
  beforeEach(function () {
    // Initialize a Medium client before each test
    client = new medium.MediumClient({ clientId: clientId, clientSecret:
clientSecret })
    nock.disableNetConnect() // Disable actual network requests
  })

  // Cleanup after each test
```

```
  afterEach(function () {
    nock.enableNetConnect() // Re-enable network requests
    delete client // Clean up client object
  })

  // Test suite for setAccessToken method
  describe('#setAccessToken', function () {

    // Test: Should set the access token correctly
    it ('sets the access token', function (done) {
      var token = 12345 // Comparison Error: Token should be a string, not a
number
      client.setAccessToken(token) // Set the token
      client._accessToken.should.be.String().and.equal(token) // Validate the
token
      done()
    })
  })

  // Test suite for getAuthorizationUrl method
  describe('#getAuthorizationUrl', function () {

    // Test: Should return a valid URL
    it ('returns a valid URL for fetching', function (done) {
      var state = "state" // Example state parameter
      var redirectUrl = "https://example.com/callback" // Example redirect URL
      var scope = [medium.Scope.BASIC_PROFILE, medium.Scope.LIST_PUBLICATIONS,
medium.Scope.PUBLISH_POST] // Example scope
      var authUrlStr = client.getAuthorizationUrl(state, redirectUrl, scope)
// Generate the authorization URL

      // Input/Output Error: url.parse() is deprecated, should use `new URL()`
instead
      var authUrl = url.parse(authUrlStr, true) // Parse the URL
      authUrl.protocol.should.equal('https:') // Check the protocol
      authUrl.hostname.should.equal('medium.com') // Check the hostname
      authUrl.pathname.should.equal('/m/oauth/authorize') // Check the path
      authUrl.query.should.deepEqual({
        client_id: clientId,
        scope: scope.join(','), // Join the scopes with commas
        response_type: 'code', // Response type
        state: state,
        redirect_uri: redirectUrl
      })
      done() // Mark the test as done
    })
  })
```

```
  // Test suite for exchangeAuthorizationCode method
  describe('#exchangeAuthorizationCode', function () {

    // Test: Should make a request and set the access token from response
    it ('makes a request for authorization_code and sets the access token from response', function (done) {
      var code = '12345' // Authorization code
      var grantType = 'authorization_code' // Grant type
      var redirectUrl = 'https://example.com/callback' // Redirect URL

      // Data Declaration Error: `requestBody` is not declared properly using `var`
      requestBody = qs.stringify({ // Missing `var` for requestBody declaration
        code: code,
        client_id: clientId,
        client_secret: clientSecret,
        grant_type: grantType,
        redirect_uri: redirectUrl
      })
      var accessToken = 'abcdef' // Mocked access token
      var refreshToken = 'ghijkl' // Mocked refresh token
      var responseBody = { access_token: accessToken, refresh_token: refreshToken }

      // Mock the request to Medium API
      var request = nock('https://api.medium.com/', {
          'Content-Type': 'application/x-www-form-urlencoded'
        })
        .post('/v1/tokens', requestBody) // Control Flow Error: Incorrect API path
        .reply(201, responseBody)

      // Exchange the authorization code for an access token
      client.exchangeAuthorizationCode(code, redirectUrl, function (err, data) {
        if (err) throw err
        // Comparison Error: Comparing token to a number (should compare to a string)
        data.access_token.should.equal(123456) // Incorrect comparison to number
        done()
      })
      request.done() // Ensure the mock request is completed
    })
  })
```

```
  // Test suite for getUser method
  describe('#getUser', function () {
    // Test: Should return user info from expected URL
    it ('gets the information from expected URL and returns contents of data
envelope', function (done) {
      var response = { data: 'response data' } // Mocked API response

      // Mock request to fetch user info
      var request = nock('https://api.medium.com')
        .get('/v1/me')
        .reply(400, response) // Interface Error: Should handle error response
gracefully

      // Fetch user info
      client.getUser(function (err, data) {
        if (err) throw err // Error should be handled instead of throwing
        data.should.deepEqual(response['data']) // Check the response
        done()
      })
      request.done() // Ensure the mock request is completed
    })
  })

  // Test suite for createPost method
  describe('#createPost', function () {

    // Test: Should make a POST request to Medium API
    it ('makes a proper POST request to the Medium API and returns contents of
data envelope', function (done) {
      var options = {
        userId: '123456', // Mock userId
        title: 'new post title', // Post title
        content: '<h1>New Post!</h1>', // Post content
        contentFormat: 'html', // Content format
        tags: ['js', 'unit tests'], // Tags
        canonicalUrl: 'http://example.com/new-post', // Canonical URL
        publishedAt: '2004-02-12T15:19:21+00:00', // Published at date
        publishStatus: 'draft', // Publish status
        license: 'all-rights-reserved' // License
      }
      var response = { data: 'response data' } // Mocked API response
      var request = nock('https://api.medium.com/')
        .post('/v1/users/' + options.userId + '/posts', {
            title: options.title,
            content: options.content,
            contentFormat: options.contentFormat,
```

```
            tags: options.tags,
            canonicalUrl: options.canonicalUrl,
            publishedAt: options.publishedAt,
            publishStatus: options.publishStatus,
            license: options.license
        })
        .reply(200, response)

      // Input/Output Error: No validation for required fields in the input
object
      client.createPost({}, function (err, data) {
        if (err) throw err // Proper error handling is required
        data.should.deepEqual(response['data']) // Check the response
        done() // Mark the test as done
      })
      request.done() // Ensure the mock request is completed
    })
  })
})
```

Exercise:

Q1. How many errors are there in the program?
A1. There are 5 key errors like

1. Comparison Errors: Incorrect type of comparison. Number was compared with the string.
2. Data Declaration Errors: Variables not declared properly
3. Control-Flow Errors: Code was running after done() is executed
4. Input-Output Errors: Incorrect method for input and output. url.parse() instead of new URL
5. Interface Errors: Error response not entered properly

Q2. Which category of program inspection would you find more effective?
A2. Static method is more useful because there are many syntax errors which can be easily corrected without running the code.

Q3. Which type of error are you not able to identify using the program inspection?
A3. Logical errors and computational errors.

Q4. Is the program inspection technique worth applying?
A4. Yes, because it resolves the error as soon as possible.

CODE 2:
Github Link: https://github.com/Medium/medium-sdk-nodejs/blob/master/lib/mediumClient.js

```
var https = require('https')
var qs = require('querystring')
```

```
var url = require('url')
var util = require('util')


var DEFAULT_ERROR_CODE = -1
var DEFAULT_TIMEOUT_MS = 5000


/*
 * Valid scope options.
 * @enum {string}
 */
var Scope = {
  BASIC_PROFILE: 'basicProfile',
  LIST_PUBLICATIONS: 'listPublications',
  PUBLISH_POST: 'publishPost'
}


/**
 * The publish status when creating a post.
 * @enum {string}
 */
var PostPublishStatus = {
  DRAFT: 'draft',
  UNLISTED: 'unlisted',
  PUBLIC: 'public'
}


/**
 * The content format to use when creating a post.
 * @enum {string}
 */
var PostContentFormat = {
  HTML: 'html',
  MARKDOWN: 'markdown'
}


/**
 * The license to use when creating a post.
 * @enum {string}
 */
var PostLicense = {
  ALL_RIGHTS_RESERVED: 'all-rights-reserved',
  CC_40_BY: 'cc-40-by',
```

```
  CC_40_BY_ND: 'cc-40-by-nd',
  CC_40_BY_SA: 'cc-40-by-sa',
  CC_40_BY_NC: 'cc-40-by-nc',
  CC_40_BY_NC_ND: 'cc-40-by-nc-nd',
  CC_40_BY_NC_SA: 'cc-40-by-nc-sa',
  CC_40_ZERO: 'cc-40-zero',
  PUBLIC_DOMAIN: 'public-domain'
}


/**
 * An error with a code.
 *
 * @param {string} message
 * @param {number} code
 * @constructor
 */
function MediumError(message, code) {
  this.message = message
  this.code = code
}
util.inherits(MediumError, Error)


/**
 * The core client.
 *
 * @param {{
 *  clientId: string,
 *  clientSecret: string
 * }} options
 * @constructor
 */
function MediumClient(options) {
  this._enforce(options, ['clientId', 'clientSecret'])
  this._clientId = options.clientId
  this._clientSecret = options.clientSecret
  this._accessToken = ""
}


/**
 * Sets an access token on the client used for making requests.
 *
 * @param {string} accessToken
 * @return {MediumClient}
 */
```

```
MediumClient.prototype.setAccessToken = function (accessToken) {
  this._accessToken = accessToken
  return this
}


/**
 * Builds a URL at which you may request authorization from the user.
 *
 * @param {string} state
 * @param {string} redirectUrl
 * @param {Array.<Scope>} requestedScope
 * @return {string}
 */
MediumClient.prototype.getAuthorizationUrl = function (state, redirectUrl,
requestedScope) {
  return url.format({
    protocol: 'https',
    host: 'medium.com',
    pathname: '/m/oauth/authorize',
    query: {
      client_id: this._clientId,
      scope: requestedScope.join(','),
      response_type: 'code',
      state: state,
      redirect_uri: redirectUrl
    }
  })
}


/**
 * Exchanges an authorization code for an access token and a refresh token.
 *
 * @param {string} code
 * @param {string} redirectUrl
 * @param {NodeCallback} callback
 */
MediumClient.prototype.exchangeAuthorizationCode = function (code,
redirectUrl, callback) {
  this._acquireAccessToken({
    code: code,
    client_id: this._clientId,
    client_secret: this._clientSecret,
    grant_type: 'authorization_code',
    redirect_uri: redirectUrl
  }, callback)
```

```
}


/**
 * Exchanges a refresh token for an access token and a refresh token.
 *
 * @param {string} refreshToken
 * @param {NodeCallback} callback
 */
MediumClient.prototype.exchangeRefreshToken = function (refreshToken,
callback) {
  this._acquireAccessToken({
    refresh_token: refreshToken,
    client_id: this._clientId,
    client_secret: this._clientSecret,
    grant_type: 'refresh_token'
  }, callback)
}


/**
 * Returns the details of the user associated with the current
 * access token.
 *
 * Requires the current access token to have the basicProfile scope.
 *
 * @param {NodeCallback} callback
 */
MediumClient.prototype.getUser = function (callback) {
  this._makeRequest({
    method: 'GET',
    path: '/v1/me'
  }, callback)
}


/**
 * Returns the publications related to the current user. Notice that
 * the userId needs to be passed in as an option. It can be acquired
 * with a call to getUser().
 *
 * Requires the current access token to have the
 * listPublications scope.
 *
 * @param {{
 *  userId: string
 * }} options
```

```
 * @param {NodeCallback} callback
 */
MediumClient.prototype.getPublicationsForUser = function (options, callback) {
  this._enforce(options, ['userId'])
  this._makeRequest({
    method: 'GET',
    path: '/v1/users/' + options.userId + '/publications'
  }, callback)
}


/**
 * Returns the contributors for a chosen publication. The publication is
identified
 * by the publication ID included in the options argument. IDs for
publications
 * can be acquired by getUsersPublications.
 *
 * Requires the current access token to have the basicProfile scope.
 *
 * @param {{
 *  publicationId: string
 * }} options
 * @param {NodeCallback} callback
 */
MediumClient.prototype.getContributorsForPublication = function (options,
callback) {
  this._enforce(options, ['publicationId'])
  this._makeRequest({
    method: 'GET',
    path: '/v1/publications/' + options.publicationId + '/contributors'
  }, callback)
}


/**
 * Creates a post on Medium.
 *
 * Requires the current access token to have the publishPost scope.
 *
 * @param {{
 *  userId: string,
 *  title: string,
 *  contentFormat: PostContentFormat,
 *  content: string,
 *  tags: Array.<string>,
 *  canonicalUrl: string,
```

```
 *  publishStatus: PostPublishStatus,
 *  license: PostLicense
 * }} options
 * @param {NodeCallback} callback
 */
MediumClient.prototype.createPost = function (options, callback) {
  this._enforce(options, ['userId'])
  this._makeRequest({
    method: 'POST',
    path: '/v1/users/' + options.userId + '/posts',
    data: {
      title: options.title,
      content: options.content,
      contentFormat: options.contentFormat,
      tags: options.tags,
      canonicalUrl: options.canonicalUrl,
      publishedAt: options.publishedAt,
      publishStatus: options.publishStatus,
      license: options.license
    }
  }, callback)
}


/**
 * Creates a post on Medium and places it under specified publication.
 * Please refer to the API documentation for rules around publishing in
 * a publication: https://github.com/Medium/medium-api-docs
 *
 * Requires the current access token to have the publishPost scope.
 *
 * @param {{
 *  userId: string,
 *  publicationId: string,
 *  title: string,
 *  contentFormat: PostContentFormat,
 *  content: string,
 *  tags: Array.<string>,
 *  canonicalUrl: string,
 *  publishStatus: PostPublishStatus,
 *  license: PostLicense
 * }} options
 * @param {NodeCallback} callback
 */
MediumClient.prototype.createPostInPublication = function (options, callback)
{
  this._enforce(options, ['publicationId'])
```

```
  this._makeRequest({
    method: 'POST',
    path: '/v1/publications/' + options.publicationId + '/posts',
    data: {
      title: options.title,
      content: options.content,
      contentFormat: options.contentFormat,
      tags: options.tags,
      canonicalUrl: options.canonicalUrl,
      publishedAt: options.publishedAt,
      publishStatus: options.publishStatus,
      license: options.license
    }
  }, callback)
}


/**
 * Acquires an access token for the Medium API.
 *
 * Sets the access token on the client on success.
 *
 * @param {Object} params
 * @param {NodeCallback} callback
 */
MediumClient.prototype._acquireAccessToken = function (params, callback) {
  this._makeRequest({
    method: 'POST',
    path: '/v1/tokens',
    contentType: 'application/x-www-form-urlencoded',
    data: qs.stringify(params)
  }, function (err, data) {
    if (!err) {
      this._accessToken = data.access_token
    }
    callback(err, data)
  }.bind(this))
}


/**
 * Enforces that given options object (first param) defines
 * all keys requested (second param). Raises an error if any
 * is missing.
 *
 * @param {Object} options
 * @param {keys} requiredKeys
```

```
 */
MediumClient.prototype._enforce = function (options, requiredKeys) {
  if (!options) {
    throw new MediumError('Parameters for this call are undefined',
DEFAULT_ERROR_CODE)
  }
  requiredKeys.forEach(function (requiredKey) {
    if (!options[requiredKey]) throw new MediumError('Missing required
parameter "' + requiredKey + '"', DEFAULT_ERROR_CODE)
  })
}



/**
 * Makes a request to the Medium API.
 *
 * @param {Object} options
 * @param {NodeCallback} callback
 */
MediumClient.prototype._makeRequest = function (options, callback) {
  var requestParams = {
    host: 'api.medium.com',
    port: 443,
    method: options.method,
    path: options.path
  }
  var req = https.request(requestParams, function (res) {
    var body = []

    res.setEncoding('utf-8')
    res.on('data', function (data) {
      body.push(data)
    })
    res.on('end', function () {
      var payload
      var responseText = body.join('')
      try {
        payload = JSON.parse(responseText)
      } catch (err) {
        callback(new MediumError('Failed to parse response',
DEFAULT_ERROR_CODE), null)
        return
      }

      var statusCode = res.statusCode
      var statusType = Math.floor(res.statusCode / 100)
```

```
      if (statusType == 4 || statusType == 5) {
        var err = payload.errors[0]
        callback(new MediumError(err.message, err.code), null)
      } else if (statusType == 2) {
        callback(null, payload.data || payload)
      } else {
        callback(new MediumError('Unexpected response', DEFAULT_ERROR_CODE),
null)
      }
    })
  }).on('error', function (err) {
    callback(new MediumError(err.message, DEFAULT_ERROR_CODE), null)
  })

  req.setHeader('Content-Type', options.contentType || 'application/json')
  req.setHeader('Authorization', 'Bearer ' + this._accessToken)
  req.setHeader('Accept', 'application/json')
  req.setHeader('Accept-Charset', 'utf-8')

  req.setTimeout(DEFAULT_TIMEOUT_MS, function () {
    // Aborting a request triggers the 'error' event.
    req.abort()
  })

  if (options.data) {
    var data = options.data
    if (typeof data == 'object') {
      data = JSON.stringify(data)
    }
    req.write(data)
  }
  req.end()
}

// Exports

module.exports = {
  MediumClient: MediumClient,
  MediumError: MediumError,
  Scope: Scope,
  PostPublishStatus: PostPublishStatus,
  PostLicense: PostLicense,
  PostContentFormat: PostContentFormat
}
```

Exercise:

Q1. How many errors are there in the program?
A1. There are 5 key errors like

1. **Callback Issue**: In the `_acquireAccessToken` method, there is a callback usage where `this._accessToken = data.access_token` could fail if `data` does not contain the `access_token` key.
2. **Error Handling Issue**: In the `_makeRequest` method, the error handling relies on `payload.errors[0]`. This could fail if `payload.errors` is undefined or not an array.
3. **Header Setting**: The method `req.setHeader` should use `req.setHeader()` instead of `req.setHeader`. It's a typo issue.
4. **Missing Options Check**: In the `createPostInPublication` method, the code checks for `publicationId` but not other required options like `userId`.
5. **No Retry Logic**: There's no retry mechanism for network or timeout errors, which could be useful when making HTTPS requests.

Q2. Which category of program inspection would you find more effective?
A2. Static method is more useful because there are many syntax errors which can be easily corrected without running the code.

Q3. Which type of error are you not able to identify using the program inspection?
A3. Runtime errors such as network issues and API responses, that work in real time.

Q4. Is the program inspection technique worth applying?
A4. Yes, because it resolves the error as soon as possible and prevents runtime errors.

CODE 3:

```
// Import required modules

var fs = require("fs") // Module to interact with the file system
var CSS = require("css") // Module to parse and stringify CSS
var path = require("path") // Module to work with file paths
var async = require("async") // Module to handle asynchronous operations

// Export the SUS module
module.exports = SUS

// SUS constructor function
function SUS (source, options) {
  if (!(this instanceof SUS)) return new SUS(source, options)
  this.source = source // CSS source code
  this.options = options || {} // Options for processing
}
```

```
// Regular expressions used for various pattern matching
SUS.DOT_REGEXP = /^./ // Matches a dot character
SUS.URL_REGEXP = /url\s*\(['"]?([^\)'"]+)['"]?\)/ // Matches URL references in
CSS
SUS.URL_REGEXP_GLOBAL = /url\s*\(['"]?[^\)'"]+['"]?\)/g // Matches all URL
references globally
SUS.PROTOCOCAL_REGEXP = /\/\// // Matches protocol (e.g., http:// or https://)

// Helper function to extend an object with additional properties
function extend (obj) {
  Array.prototype.slice.call(arguments, 1).forEach(function (source) {
    for (var prop in source) {
      obj[prop] = source[prop] // Copy properties from source to obj
    }
  })
  return obj // Return the extended object
}

// Function to parse CSS rules and extract URL references for sprite inlining
function parseRules (base, sprites, options, complete) {

  // Filter through CSS rules
  async.filterSeries(base.rules, function (rule, nextRule) {

    // If the rule contains nested rules (e.g., media queries), handle them
recursively
    if (rule.rules) {
      var spriteKey = sprites.rules.length
      var _sprite = extend({}, rule)
      _sprite.rules = []
      sprites.rules.push(_sprite)

      // Recursively parse the nested rules
      return parseRules(rule, _sprite, options, function (err, result) {
        if (!_sprite.rules.length) sprites.rules.splice(spriteKey, 1)
        nextRule(rule.rules = result.length && result)
      })
    }

    // Process the rule asynchronously to avoid callstack overflow
    setImmediate(function () {

      // Skip keyframe and @font-face declarations, as we don't need to
process those
      if (rule.keyframes || (rule.selectors[0] && rule.selectors[0] ==
'@font-face')) {
        return nextRule(rule)
```

```
      }

      // Filter declarations that contain URL references
      async.filterSeries(rule.declarations, function (declaration,
nextDeclaration) {
        var files = declaration.value.match(SUS.URL_REGEXP_GLOBAL)

        // If no URLs are found, skip this declaration
        if (!files) return nextDeclaration(declaration)

        // Process all matched URL references in the declaration
        async.map(files, function (file, nextFile) {
          var filepath = file.match(SUS.URL_REGEXP)[1]

          // Skip URLs that are not local (i.e., remote URLs)
          if (SUS.PROTOCOCAL_REGEXP.test(filepath)) return nextFile(null)

          // Resolve the filepath based on the base option (if provided)
          if (typeof options.base == 'function') {
            filepath = options.base(filepath)
          } else if (typeof options.base != 'undefined') {
            filepath = path.join(options.base, filepath)
          }

          // Read the image file and convert it to base64 for inline use
          fs.readFile(filepath, { encoding: "base64" }, function (err, data) {
            if (err) return complete(err)
            nextFile(null, {
              expression: file, // The original URL in the CSS
              ext: path.extname(filepath).replace(SUS.DOT_REGEXP, ""), //
Extract file extension
              path: filepath, // File path of the image
              data: data // Base64-encoded image data
            })
          })

        // Once all files are processed, replace URLs with base64 data URIs
        }, function (err, results) {
          if (!results.filter(function (r) { return r }).length) return
nextDeclaration(declaration)
          parseSprite(results, sprites, rule, declaration, nextDeclaration)
        })

      }, function (result) {
        nextRule(rule.declarations = result.length && result)
      })
```

```
    })

  }, function (result) {
    complete(null, (base.rules = result))
  })
}

// Function to parse and inline sprite images into CSS rules
function parseSprite(files, sprites, rule, declaration, complete) {
  var value = declaration.value
  var spriteRule
  var spriteDeclaration
  var dataURI

  // Replace all URLs with base64-encoded data URIs
  files.forEach(function (file) {
    dataURI = "url(" + file.data + ")"
    value = value.replace(file.expression, dataURI)
  })

  // Define a new CSS declaration for the inlined sprite
  spriteDeclaration = {
    "property": declaration.property,
    "value": value
  }

  // Define a new CSS rule with the inlined sprite
  spriteRule = {
    "selectors": rule.selectors,
    "declarations": [ spriteDeclaration ]
  }

  // Remove the original declaration from the base CSS
  declaration = null

  // Add the new sprite rule to the sprites CSS object
  sprites.rules.push(spriteRule)

  complete(declaration)
}

// Method to parse the CSS source and extract sprites
SUS.prototype.parse = function (callback) {
  this._base = CSS.parse(this.source) // Parse the CSS source
  this._sprites = { "stylesheet": { "rules": [] } } // Initialize sprites
object
```

```
  // Parse rules and return the result in the callback
  parseRules(this._base.stylesheet, this._sprites.stylesheet, this.options,
function (err) {
    callback(err, this)
  }.bind(this))

  return this
}

// Method to return the base CSS as a string
SUS.prototype.base = function () {
  return CSS.stringify(this._base) // Stringify the base CSS object
}

// Method to return the sprites CSS as a string
SUS.prototype.sprites = function () {
  return CSS.stringify(this._sprites) // Stringify the sprites CSS object
}
```

Exercise:

Q1. How many errors are there in the program?
A1. **Encoding Issue in `fs.readFile`**: The second argument `"base64"` in `fs.readFile(filepath, "base64", ...)` is incorrect. The correct way to read a file as base64 is to use `fs.readFile(filepath, { encoding: "base64" }, ...)` instead of passing a string.
Potential Misuse of `setImmediate`: Although `setImmediate` is used to prevent stack overflow, it may cause unnecessary delays in code execution. It is not an error, but a potential inefficiency.
Incomplete Callback Management: In the `parseSprite` function, `declaration = null` might lead to issues when modifying the same object passed by reference. This should be handled with caution, as mutating objects directly could lead to unintended results.
Redundant Checks: The line `if (!files) return nextDeclaration(declaration)` in `parseRules` is checking for URLs in a CSS declaration. If `a file` is empty or not present, it returns the same declaration, which is an unnecessary redundancy.
Misuse of `this` in Callback: In `SUS.prototype.parse`, `this` is being used inside the `parseRules` callback, but since the callback is asynchronous, `this` might not refer to the correct context. It should be handled with `.bind(this)` to maintain scope.

Q2. Which category of program inspection would you find more effective?
A2. Static because there is redundancy in the code and syntax errors, which are easily identified by static testing.

Q3. Which type of error are you not able to identify using the program inspection?
A3. Runtime errors such as network issues and if there are any incorrect file paths then they

are not identified just by reading the code.

Q4. Is the program inspection technique worth applying?
A4. Yes, because it resolves the error as soon as possible and prevents runtime errors.

CODE 4:
Github Link: https://github.com/SahilM2063/Simple-Chess-Using-Javascript/blob/main/app.js

```
const gameBoard = document.querySelector("#gameboard");
const playerDetails = document.querySelector("#player");
const infoDisplay = document.querySelector("#info-display");
const err = document.querySeconst gameBoard =
document.querySelector("#gameboard");
const playerDetails = document.querySelector("#player");
const infoDisplay = document.querySelector("#info-display");
const err = document.querySelector("#err");
const width = 8

let playerTurn = 'black';
playerDetails.textContent = 'black'

const startPieces = [
    Rook, Knight, Bishop, Queen, King, Bishop, Knight, Rook,
    Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn,
    '', '', '', '', '', '', '', '',
    '', '', '', '', '', '', '', '',
    '', '', '', '', '', '', '', '',
    '', '', '', '', '', '', '', '',
    Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn,
    Rook, Knight, Bishop, King, Queen, Bishop, Knight, Rook,
]

function createBoard() {
    startPieces.forEach((startPiece, i) => {
        const square = document.createElement("div");
        square.classList.add("square");
        square.innerHTML = startPiece

        square.setAttribute("square-id", i);
        square.firstChild?.setAttribute('draggable', true)

        const row = Math.floor((63 - i) / 8) + 1;

        if (row % 2 === 0) {
            square.classList.add(i % 2 == 0 ? "beige" : "brown");
        } else {
```

```
            square.classList.add(i % 2 == 0 ? "brown" : "beige");
        }

        if (i <= 15) {
            square.firstChild.firstChild.classList.add("black");
        }
        if (i >= 48) {
            square.firstChild.firstChild.classList.add("white");
        }


        gameBoard.append(square);
    });
};

createBoard();


const allSquares = document.querySelectorAll("#gameboard .square");
// console.log(allSquares)

allSquares.forEach(square => {
    square.addEventListener('dragstart', dragstart);
    square.addEventListener('dragover', dragover);
    square.addEventListener('drop', dragdrop);
})

let startPositionId
let draggedElement

function dragstart(e) {
    startPositionId = e.target.parentNode.getAttribute("square-id")
    draggedElement = e.target
}

function dragover(e) {
    e.preventDefault();
}

function dragdrop(e) {
    e.stopPropagation();

    // console.log('player go', playerTurn)
    // console.log('target', e.target)
    // console.log(draggedElement)

    const correctTurn =
```

```
draggedElement.firstChild.classList.contains(playerTurn);
    const taken = e.target.classList.contains('piece');
    const valid = checkIfValid(e.target);
    const opponentTurn = playerTurn === 'white' ? 'black' : 'white';
    const takenByOpponent =
e.target.firstChild?.classList.contains(opponentTurn);
    // console.log('opp go', opponentTurn)


    if (correctTurn) {
        // must check this condition
        if (takenByOpponent && valid) {
            e.target.parentNode.append(draggedElement);
            e.target.remove();
            checkForWin();
            changePlayer();
            return
        }
        if (taken && !takenByOpponent) {
            err.textContent = 'Can not go there'
            setTimeout(() => {
                err.textContent = ''
            }, 2000);
            return
        }
        if (valid) {
            e.target.append(draggedElement);
            checkForWin();
            changePlayer();
            return
        }
    }
}

function checkIfValid(target) {
    const targetId = Number(target.getAttribute('square-id')) ||
Number(target.parentNode.getAttribute('square-id'));
    const startId = Number(startPositionId);
    const piece = draggedElement.id
    console.log(startId, targetId, piece)

    switch (piece) {
        case 'pawn':
            const starterRow = [8, 9, 10, 11, 12, 13, 14, 15];
            if (starterRow.includes(startId) && startId + width * 2 ===
targetId ||
                startId + width === targetId ||
```

```
                startId + width - 1 === targetId &&
document.querySelector([square-id = "${startId + width - 1}"]).firstChild ||
                startId + width + 1 === targetId &&
document.querySelector([square-id = "${startId + width + 1}"]).firstChild) {
                return true;
            }
            break;
        case 'knight':
            if (
                startId + width * 2 + 1 === targetId ||
                startId + width * 2 - 1 === targetId ||
                startId + width - 2 === targetId ||
                startId + width + 2 === targetId ||
                startId - width * 2 + 1 === targetId ||
                startId - width * 2 - 1 === targetId ||
                startId - width + 2 === targetId ||
                startId - width - 2 === targetId
            ) {
                return true
            }
            break;

        case 'bishop':
            if (
                // for right cross --- forward
                startId + width + 1 === targetId ||
                startId + width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild ||
                startId + width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
||
                startId + width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild ||
                startId + width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild ||
                startId + width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
```

```
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild ||
                startId + width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild ||

                // for left cross --- forward
                startId + width - 1 === targetId ||
                startId + width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild ||
                startId + width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
||
                startId + width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild ||
                startId + width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild ||
                startId + width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild ||
                startId + width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||
```

```
                // for right cross --- backward
                startId - width - 1 === targetId ||
                startId - width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild ||
                startId - width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
||
                startId - width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild ||
                startId - width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild ||
                startId - width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild ||
                startId - width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // for left cross -- backward
                startId - width + 1 === targetId ||
                startId - width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild ||
                startId - width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
||
                startId - width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
```

```
3}"]).firstChild ||
                startId - width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild ||
                startId - width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild ||
                startId - width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild
            ) {
                return true;
            }
            break;

        case 'rook':
            if (
                // moving straight forward
                startId + width === targetId ||
                startId + width * 2 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild ||
                startId + width * 3 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild ||
                startId + width * 4 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild ||
                startId + width * 5 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild ||
                startId + width * 6 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
```

```
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild ||
                startId + width * 7 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 6}"]).firstChild ||

                // moving straight backward
                startId - width === targetId ||
                startId - width * 2 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild ||
                startId - width * 3 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild ||
                startId - width * 4 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild ||
                startId - width * 5 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild ||
                startId - width * 6 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild ||
                startId - width * 7 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 6}"]).firstChild ||

                // moving left side straight
                startId + 1 === targetId ||
                startId + 2 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild ||
                startId + 3 === targetId &&
```

```
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild ||
                startId + 4 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild ||
                startId + 5 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild ||
                startId + 6 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild ||
                startId + 7 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild &&
!document.querySelector([square-id="${startId + 6}"]).firstChild ||

                // moving right side straight
                startId - 1 === targetId ||
                startId - 2 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild ||
                startId - 3 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild ||
                startId - 4 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild ||
                startId - 5 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild ||
                startId - 6 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild ||
```

```
                startId - 7 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild &&
!document.querySelector([square-id="${startId - 6}"]).firstChild
            ) {
                return true
            }
            break;

        case 'queen':
            if (
                // for right cross --- forward
                startId + width + 1 === targetId ||
                startId + width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild ||
                startId + width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
||
                startId + width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild ||
                startId + width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild ||
                startId + width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild ||
                startId + width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
```

```
"${startId + width * 6 + 6}"]).firstChild ||

                // for left cross --- forward
                startId + width - 1 === targetId ||
                startId + width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild ||
                startId + width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
||
                startId + width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild ||
                startId + width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild ||
                startId + width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild ||
                startId + width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // for right cross --- backward
                startId - width - 1 === targetId ||
                startId - width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild ||
                startId - width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
||
                startId - width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
```

```
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild ||
                startId - width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild ||
                startId - width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild ||
                startId - width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // fot left cross -- backward
                startId - width + 1 === targetId ||
                startId - width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild ||
                startId - width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
||
                startId - width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild ||
                startId - width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild ||
                startId - width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
```

```
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild ||
                startId - width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild ||

                // moving straight forward
                startId + width === targetId ||
                startId + width * 2 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild ||
                startId + width * 3 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild ||
                startId + width * 4 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild ||
                startId + width * 5 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild ||
                startId + width * 6 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild ||
                startId + width * 7 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 6}"]).firstChild ||

                // moving straight backward
                startId - width === targetId ||
                startId - width * 2 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild ||
```

```
                startId - width * 3 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild ||
                startId - width * 4 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild ||
                startId - width * 5 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild ||
                startId - width * 6 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild ||
                startId - width * 7 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 6}"]).firstChild ||

                // moving left side straight
                startId + 1 === targetId ||
                startId + 2 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild ||
                startId + 3 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild ||
                startId + 4 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild ||
                startId + 5 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild ||
                startId + 6 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
```

```
!document.querySelector([square-id="${startId + 5}"]).firstChild ||
                startId + 7 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild &&
!document.querySelector([square-id="${startId + 6}"]).firstChild ||

                // moving right side straight
                startId - 1 === targetId ||
                startId - 2 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild ||
                startId - 3 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild ||
                startId - 4 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild ||
                startId - 5 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild ||
                startId - 6 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild ||
                startId - 7 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild &&
!document.querySelector([square-id="${startId - 6}"]).firstChild
            ) {
                return true
            }
            break;

        case 'king':
            if (
                startId + 1 === targetId ||
                startId - 1 === targetId ||
```

```
                startId + width === targetId ||
                startId + width + 1 === targetId ||
                startId + width - 1 === targetId ||
                startId - width === targetId ||
                startId - width + 1 === targetId ||
                startId - width - 1 === targetId
            ) {
                return true
            }
            break;
        default:
            break;
    }
}


function changePlayer() {
    if (playerTurn === 'black') {
        reverseIds()
        playerTurn = 'white';
        playerDetails.textContent = 'white'
    } else {
        revertIds();
        playerTurn = 'black'
        playerDetails.textContent = 'black'
    }
}

function reverseIds() {
    const allSquares = document.querySelectorAll('#gameboard .square');
    allSquares.forEach((square, i) => {
        square.setAttribute('square-id', (width * width - 1) - i)
    })
}

function revertIds() {
    const allSquares = document.querySelectorAll('#gameboard .square');
    allSquares.forEach((square, i) => {
        square.setAttribute('square-id', i)
    })
}

function checkForWin() {
    const kings = Array.from(document.querySelectorAll('#king'));

    if (!kings.some(king => king.firstChild.classList.contains('white'))) {
        infoDisplay.innerHTML = "Black Player Wins!";
```

```
        const allSquares = document.querySelectorAll('.square');
        allSquares.forEach(square =>
square.firstChild?.setAttribute('draggable', false));
    }
    if (!kings.some(king => king.firstChild.classList.contains('black'))) {
        infoDisplay.innerHTML = "White Player Wins!";
        const allSquares = document.querySelectorAll('.square');
        allSquares.forEach(square =>
square.firstChild?.setAttribute('draggable', false));
    }
}
lector("#err");
const width = 8

let playerTurn = 'black';
playerDetails.textContent = 'black'

const startPieces = [
    Rook, Knight, Bishop, Queen, King, Bishop, Knight, Rook,
    Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn,
    '', '', '', '', '', '', '', '',
    '', '', '', '', '', '', '', '',
    '', '', '', '', '', '', '', '',
    '', '', '', '', '', '', '', '',
    Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn, Pawn,
    Rook, Knight, Bishop, King, Queen, Bishop, Knight, Rook,
]

function createBoard() {
    startPieces.forEach((startPiece, i) => {
        const square = document.createElement("div");
        square.classList.add("square");
        square.innerHTML = startPiece

        square.setAttribute("square-id", i);
        square.firstChild?.setAttribute('draggable', true)

        const row = Math.floor((63 - i) / 8) + 1;

        if (row % 2 === 0) {
            square.classList.add(i % 2 == 0 ? "beige" : "brown");
        } else {
            square.classList.add(i % 2 == 0 ? "brown" : "beige");
        }

        if (i <= 15) {
            square.firstChild.firstChild.classList.add("black");
```

```
        }
        if (i >= 48) {
            square.firstChild.firstChild.classList.add("white");
        }


        gameBoard.append(square);
    });
};

createBoard();


const allSquares = document.querySelectorAll("#gameboard .square");
// console.log(allSquares)

allSquares.forEach(square => {
    square.addEventListener('dragstart', dragstart);
    square.addEventListener('dragover', dragover);
    square.addEventListener('drop', dragdrop);
})

let startPositionId
let draggedElement

function dragstart(e) {
    startPositionId = e.target.parentNode.getAttribute("square-id")
    draggedElement = e.target
}

function dragover(e) {
    e.preventDefault();
}

function dragdrop(e) {
    e.stopPropagation();

    // console.log('player go', playerTurn)
    // console.log('target', e.target)
    // console.log(draggedElement)

    const correctTurn =
draggedElement.firstChild.classList.contains(playerTurn);
    const taken = e.target.classList.contains('piece');
    const valid = checkIfValid(e.target);
    const opponentTurn = playerTurn === 'white' ? 'black' : 'white';
    const takenByOpponent =
```

```
e.target.firstChild?.classList.contains(opponentTurn);
    // console.log('opp go', opponentTurn)


    if (correctTurn) {
        // must check this condition
        if (takenByOpponent && valid) {
            e.target.parentNode.append(draggedElement);
            e.target.remove();
            checkForWin();
            changePlayer();
            return
        }
        if (taken && !takenByOpponent) {
            err.textContent = 'Can not go there'
            setTimeout(() => {
                err.textContent = ''
            }, 2000);
            return
        }
        if (valid) {
            e.target.append(draggedElement);
            checkForWin();
            changePlayer();
            return
        }
    }
}

function checkIfValid(target) {
    const targetId = Number(target.getAttribute('square-id')) ||
Number(target.parentNode.getAttribute('square-id'));
    const startId = Number(startPositionId);
    const piece = draggedElement.id
    console.log(startId, targetId, piece)

    switch (piece) {
        case 'pawn':
            const starterRow = [8, 9, 10, 11, 12, 13, 14, 15];
            if (starterRow.includes(startId) && startId + width * 2 ===
targetId ||
                startId + width === targetId ||
                startId + width - 1 === targetId &&
document.querySelector([square-id = "${startId + width - 1}"]).firstChild ||
                startId + width + 1 === targetId &&
document.querySelector([square-id = "${startId + width + 1}"]).firstChild) {
                return true;
```

```
            }
            break;
        case 'knight':
            if (
                startId + width * 2 + 1 === targetId ||
                startId + width * 2 - 1 === targetId ||
                startId + width - 2 === targetId ||
                startId + width + 2 === targetId ||
                startId - width * 2 + 1 === targetId ||
                startId - width * 2 - 1 === targetId ||
                startId - width + 2 === targetId ||
                startId - width - 2 === targetId
            ) {
                return true
            }
            break;

        case 'bishop':
            if (
                // for right cross --- forward
                startId + width + 1 === targetId ||
                startId + width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild ||
                startId + width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
||
                startId + width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild ||
                startId + width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild ||
                startId + width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild ||
                startId + width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
```

```
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild ||

                // for left cross --- forward
                startId + width - 1 === targetId ||
                startId + width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild ||
                startId + width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
||
                startId + width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild ||
                startId + width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild ||
                startId + width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild ||
                startId + width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // for right cross --- backward
                startId - width - 1 === targetId ||
                startId - width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild ||
                startId - width * 3 - 3 === targetId &&
```

```
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
||
                startId - width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild ||
                startId - width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild ||
                startId - width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild ||
                startId - width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // fot left cross -- backward
                startId - width + 1 === targetId ||
                startId - width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild ||
                startId - width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
||
                startId - width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild ||
                startId - width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
```

```
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild ||
                startId - width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild ||
                startId - width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild
            ) {
                return true;
            }
            break;

        case 'rook':
            if (
                // moving straight forward
                startId + width === targetId ||
                startId + width * 2 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild ||
                startId + width * 3 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild ||
                startId + width * 4 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild ||
                startId + width * 5 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild ||
                startId + width * 6 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild ||
                startId + width * 7 === targetId &&
```

```
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 6}"]).firstChild ||

                // moving straight backward
                startId - width === targetId ||
                startId - width * 2 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild ||
                startId - width * 3 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild ||
                startId - width * 4 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild ||
                startId - width * 5 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild ||
                startId - width * 6 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild ||
                startId - width * 7 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 6}"]).firstChild ||

                // moving left side straight
                startId + 1 === targetId ||
                startId + 2 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild ||
                startId + 3 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild ||
                startId + 4 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
```

```
!document.querySelector([square-id="${startId + 3}"]).firstChild ||
                startId + 5 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild ||
                startId + 6 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild ||
                startId + 7 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild &&
!document.querySelector([square-id="${startId + 6}"]).firstChild ||

                // moving right side straight
                startId - 1 === targetId ||
                startId - 2 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild ||
                startId - 3 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild ||
                startId - 4 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild ||
                startId - 5 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild ||
                startId - 6 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild ||
                startId - 7 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
```

```
!document.querySelector([square-id="${startId - 5}"]).firstChild &&
!document.querySelector([square-id="${startId - 6}"]).firstChild
            ) {
                return true
            }
            break;

        case 'queen':
            if (
                // for right cross --- forward
                startId + width + 1 === targetId ||
                startId + width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild ||
                startId + width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
||
                startId + width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild ||
                startId + width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild ||
                startId + width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild ||
                startId + width * 7 + 7 === targetId &&
!document.querySelector([square-id = "${startId + width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild ||

                // for left cross --- forward
                startId + width - 1 === targetId ||
                startId + width * 2 - 2 === targetId &&
```

```
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild ||
                startId + width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
||
                startId + width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild ||
                startId + width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild ||
                startId + width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild ||
                startId + width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId + width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId + width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId + width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId + width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId +
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // for right cross --- backward
                startId - width - 1 === targetId ||
                startId - width * 2 - 2 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild ||
                startId - width * 3 - 3 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
||
                startId - width * 4 - 4 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild ||
                startId - width * 5 - 5 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
```

```
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild ||
                startId - width * 6 - 6 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild ||
                startId - width * 7 - 7 === targetId &&
!document.querySelector([square-id = "${startId - width - 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 - 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 -
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 - 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 - 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 - 6}"]).firstChild ||

                // fot left cross -- backward
                startId - width + 1 === targetId ||
                startId - width * 2 + 2 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild ||
                startId - width * 3 + 3 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
||
                startId - width * 4 + 4 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild ||
                startId - width * 5 + 5 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild ||
                startId - width * 6 + 6 === targetId &&
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild ||
                startId - width * 7 + 7 === targetId &&
```

```
!document.querySelector([square-id = "${startId - width + 1}"]).firstChild &&
!document.querySelector([square-id = "${startId - width * 2 + 2}"]).firstChild
&& !document.querySelector([square-id = "${startId - width * 3 +
3}"]).firstChild && !document.querySelector([square-id = "${startId - width *
4 + 4}"]).firstChild && !document.querySelector([square-id = "${startId -
width * 5 + 5}"]).firstChild && !document.querySelector([square-id =
"${startId + width * 6 + 6}"]).firstChild ||

                // moving straight forward
                startId + width === targetId ||
                startId + width * 2 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild ||
                startId + width * 3 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild ||
                startId + width * 4 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild ||
                startId + width * 5 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild ||
                startId + width * 6 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild ||
                startId + width * 7 === targetId &&
!document.querySelector([square-id="${startId + width}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId + width * 6}"]).firstChild ||

                // moving straight backward
                startId - width === targetId ||
                startId - width * 2 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild ||
                startId - width * 3 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild ||
                startId - width * 4 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
```

```
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild ||
                startId - width * 5 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild ||
                startId - width * 6 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild ||
                startId - width * 7 === targetId &&
!document.querySelector([square-id="${startId - width}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 2}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 3}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 4}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 5}"]).firstChild &&
!document.querySelector([square-id="${startId - width * 6}"]).firstChild ||

                // moving left side straight
                startId + 1 === targetId ||
                startId + 2 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild ||
                startId + 3 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild ||
                startId + 4 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild ||
                startId + 5 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild ||
                startId + 6 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild ||
                startId + 7 === targetId &&
!document.querySelector([square-id="${startId + 1}"]).firstChild &&
!document.querySelector([square-id="${startId + 2}"]).firstChild &&
!document.querySelector([square-id="${startId + 3}"]).firstChild &&
```

```
!document.querySelector([square-id="${startId + 4}"]).firstChild &&
!document.querySelector([square-id="${startId + 5}"]).firstChild &&
!document.querySelector([square-id="${startId + 6}"]).firstChild ||

                // moving right side straight
                startId - 1 === targetId ||
                startId - 2 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild ||
                startId - 3 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild ||
                startId - 4 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild ||
                startId - 5 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild ||
                startId - 6 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild ||
                startId - 7 === targetId &&
!document.querySelector([square-id="${startId - 1}"]).firstChild &&
!document.querySelector([square-id="${startId - 2}"]).firstChild &&
!document.querySelector([square-id="${startId - 3}"]).firstChild &&
!document.querySelector([square-id="${startId - 4}"]).firstChild &&
!document.querySelector([square-id="${startId - 5}"]).firstChild &&
!document.querySelector([square-id="${startId - 6}"]).firstChild
            ) {
                return true
            }
            break;

        case 'king':
            if (
                startId + 1 === targetId ||
                startId - 1 === targetId ||
                startId + width === targetId ||
                startId + width + 1 === targetId ||
                startId + width - 1 === targetId ||
                startId - width === targetId ||
                startId - width + 1 === targetId ||
```

```
                startId - width - 1 === targetId
            ) {
                return true
            }
            break;
        default:
            break;
    }
}


function changePlayer() {
    if (playerTurn === 'black') {
        reverseIds()
        playerTurn = 'white';
        playerDetails.textContent = 'white'
    } else {
        revertIds();
        playerTurn = 'black'
        playerDetails.textContent = 'black'
    }
}

function reverseIds() {
    const allSquares = document.querySelectorAll('#gameboard .square');
    allSquares.forEach((square, i) => {
        square.setAttribute('square-id', (width * width - 1) - i)
    })
}

function revertIds() {
    const allSquares = document.querySelectorAll('#gameboard .square');
    allSquares.forEach((square, i) => {
        square.setAttribute('square-id', i)
    })
}

function checkForWin() {
    const kings = Array.from(document.querySelectorAll('#king'));

    if (!kings.some(king => king.firstChild.classList.contains('white'))) {
        infoDisplay.innerHTML = "Black Player Wins!";
        const allSquares = document.querySelectorAll('.square');
        allSquares.forEach(square =>
square.firstChild?.setAttribute('draggable', false));
    }
    if (!kings.some(king => king.firstChild.classList.contains('black'))) {
```

```
        infoDisplay.innerHTML = "White Player Wins!";
        const allSquares = document.querySelectorAll('.square');
        allSquares.forEach(square =>
square.firstChild?.setAttribute('draggable', false));
    }
}
```

Exercise:

Q1. How many errors are there in the program?
A1. The Errors are:

1. In code add 1 in row, row = Math.floor((63-i)/8)+1, 2 time defend same variable using const, not use try catch for handling the errors.
2. Category C: Computation Errors, Category B: Data-Declaration Errors
3. Logical errors and input output errors not find using program inspection
4. YES, program inspection is worth applying as it helps in identifying potential issues early in the development process.

Q2. Which category of program inspection would you find more effective?
A2. Static because there are syntax errors, which are easily identified by static testing.

Q3. Which type of error are you not able to identify using the program inspection?
A3. Runtime errors like infinite loop,they are not identified just by reading the code.

Q4. Is the program inspection technique worth applying?
A4. Yes, because it resolves the error as soon as possible and prevents runtime errors.