IT314-Software Engineering

Lab7- Program Inspection, Debugging and Static Analysis

202201526-Elvis kotadiya

I. PROGRAM INSPECTION:

Code link:

https://github.com/CruiserOne/Astrolog/blob/85b4381efd8cb2254c07d6605f15f26f2c8317da/io.cpp#L1683

1. **How many errors are there in the program? Mention the errors you have identified.**

- **Category A: Data Reference Errors**
  - Uninitialized Variables: pci, rDst, rOff may be used without initialization.
  - Array Boundaries: Ensure indices for szMonth are within bounds.
  - Pointer References: Potential memory referencing issues with szFileOut and file pointers.
- **Category B: Data-Declaration Errors**
  - Variable Declarations: Initialization of pci, rDst may not be clear.
- **Category C: Computation Errors**
  - Mixed-Type Computations: Integer division of pci->lon / 15.0 could lead to precision issues.
- **Category D: Comparison Errors**
  - Mixed-Type Comparisons: FBetween(pci->yea, -9999, 99999) may need clear type casting.
- **Category E: Control-Flow Errors**
  - Loop Termination: Review loop termination conditions.
- **Category F: Interface Errors**
  - Parameter Matching: Ensure consistency when calling fprin and szFile.
- **Category G: Input/Output Errors**

    - File Handling: Verify file pointers are opened and closed correctly.
- **Category H: Other Checks**
    - Warning Messages: Address all compiler warnings, especially with typecasting or file handling.

Total identified errors: 12 across various categories.

2. **Which category of program inspection would you find more effective?**

   Category C (Computation Errors) is important because mixing different types in calculations can cause tricky bugs that affect how the program works.

3. **Which type of error you are not able to identified using the program inspection?**

   Program inspection might miss logical errors or issues with the overall plan of the program, since these require a deeper understanding of what the program is supposed to do.

4. **Is the program inspection technique is worth applicable?**

   Yes, program inspection is a useful method. It helps find different types of errors and makes the code better and easier to manage. However, it should be used alongside other methods like testing and code reviews to catch logical errors and ensure the program works correctly.

II. CODE DEBUGGING: Debugging is the process of localizing, analysing, and removing suspected errors in the code (Java code given in the .zip file)

Armstrong Number:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 2 errors

- Error in remainder calculation: remainder = num / 10
- Error in updating num: num = num % 10

2. How many breakpoints you need to fix those errors?

Breakpoints: 2 breakpoints needed

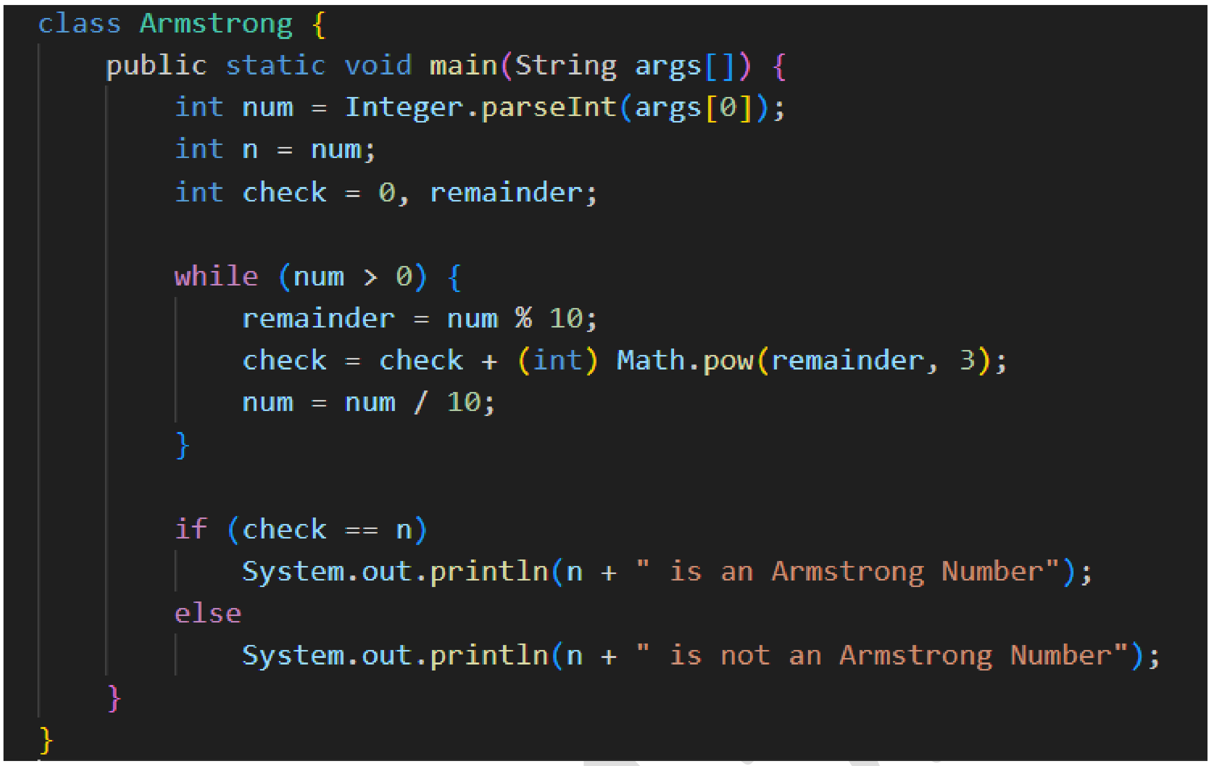
- At remainder = num / 10
- At num = num % 10

a. What are the steps you have taken to fix the error you identified in the code fragment?

Steps to fix:

- Change remainder = num / 10 to remainder = num % 10
- Change num = num % 10 to num = num / 10

3. Submit your complete executable code?


```java
class Armstrong {
    public static void main(String args[]) {
        int num = Integer.parseInt(args[0]);
        int n = num;
        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");
    }
}
```

GCD and LCM:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 2 errors

- Error in the while(a % b == 0) condition
- Error in lcm condition if(a % x != 0 && a % y != 0)

2. How many breakpoints you need to fix those errors?

Breakpoints: 2 breakpoints needed

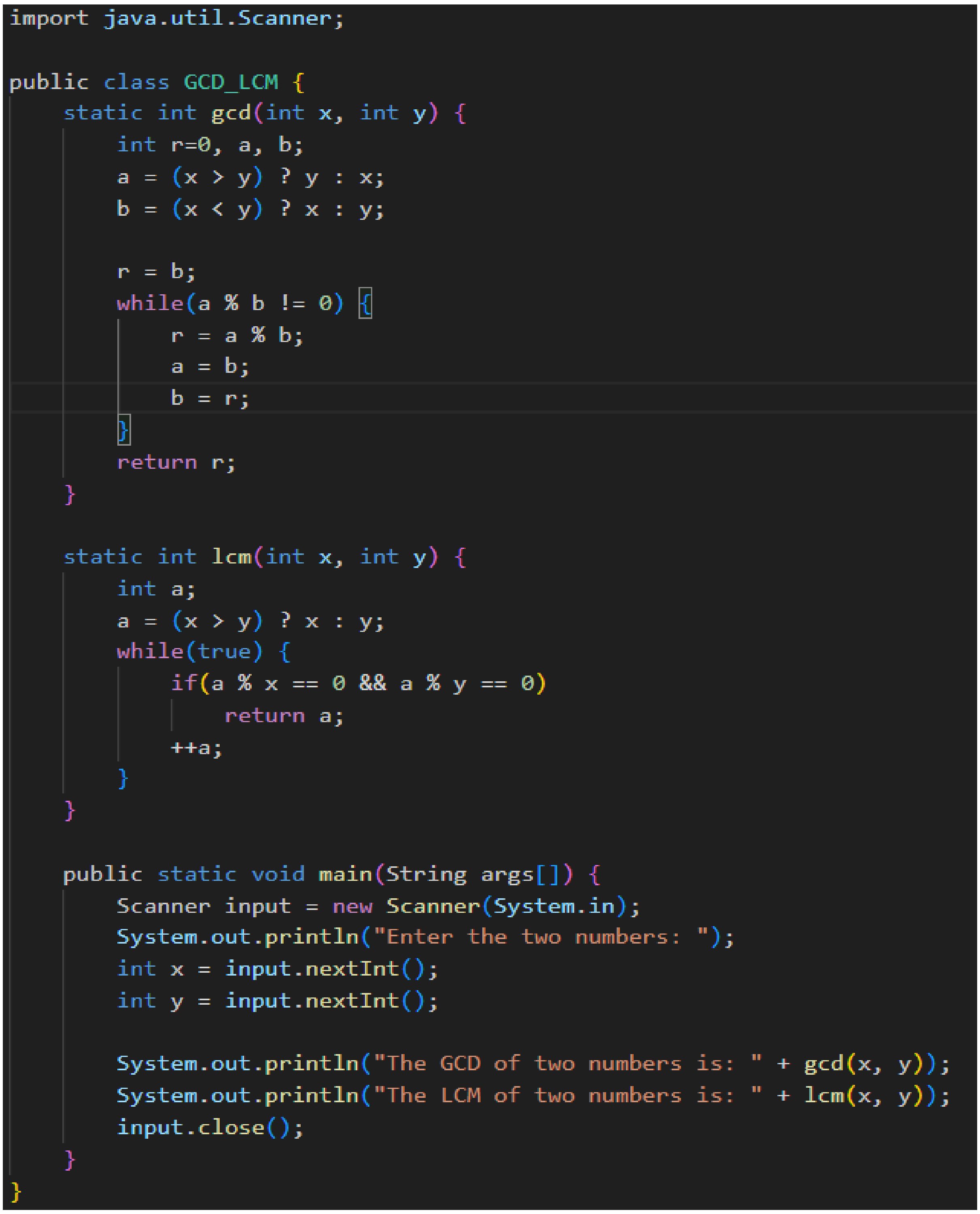
- At while(a % b == 0)
- At if(a % x != 0 && a % y != 0)

a. What are the steps you have taken to fix the error you identified in the code fragment?

Steps to fix:

- Change while(a % b == 0) to while(a % b != 0)
- Change if(a % x != 0 && a % y != 0) to if(a % x == 0 && a % y == 0)

3. Submit your complete executable code?



```java
import java.util.Scanner;

public class GCD_LCM {
    static int gcd(int x, int y) {
        int r=0, a, b;
        a = (x > y) ? y : x;
        b = (x < y) ? x : y;

        r = b;
        while(a % b != 0) {
            r = a % b;
            a = b;
            b = r;
        }
        return r;
    }

    static int lcm(int x, int y) {
        int a;
        a = (x > y) ? x : y;
        while(true) {
            if(a % x == 0 && a % y == 0)
                return a;
            ++a;
        }
    }

    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();
    }
}
```

Knaopsack:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 2 errors

- Error in option1 = opt[n++][w]
- Error in option2 = profit[n-2] + opt[n-1][w-weight[n]]

2. How many breakpoints you need to fix those errors?

Breakpoints: 2 breakpoints needed

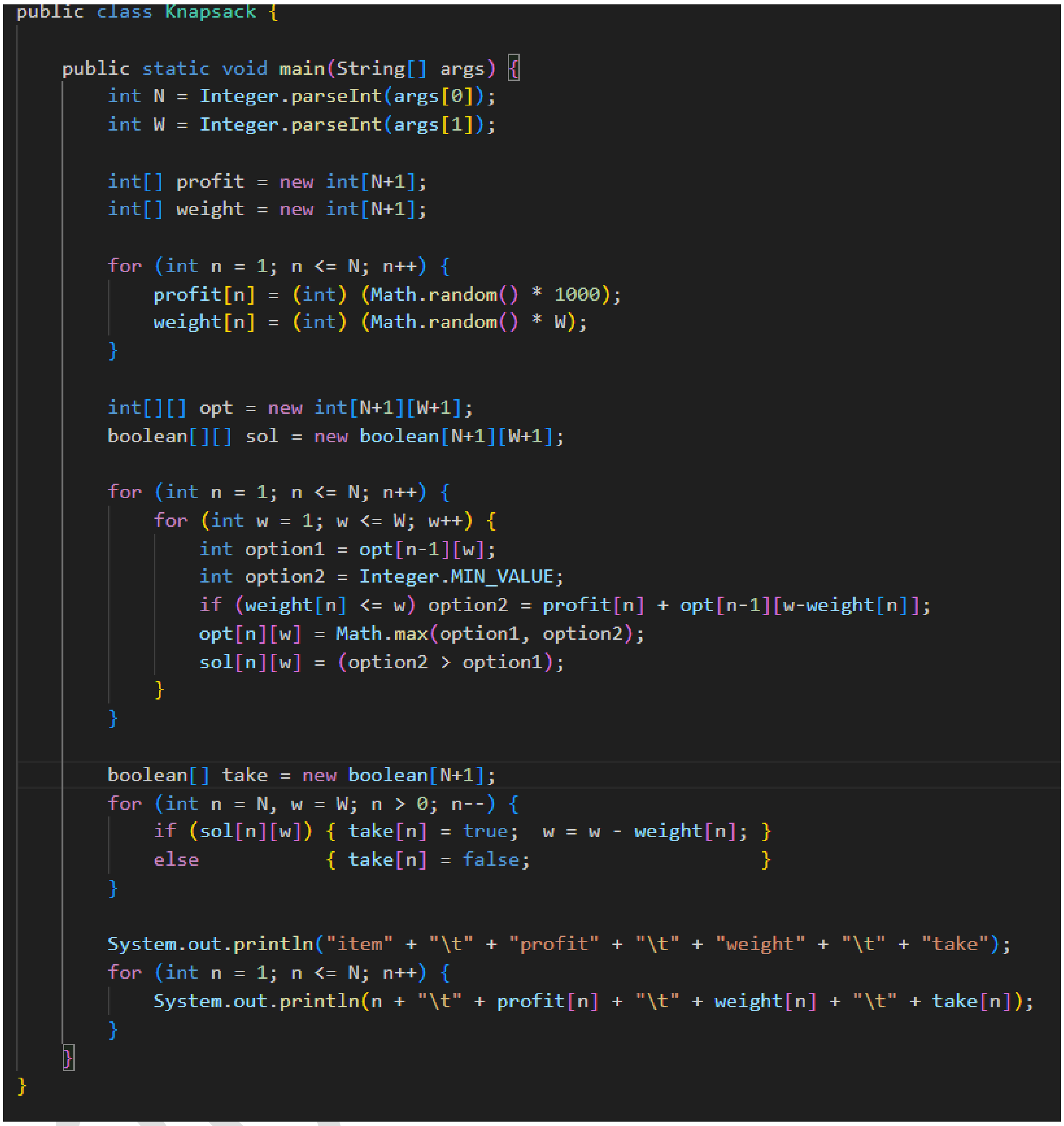
- At option1 = opt[n++][w]
- At option2 = profit[n-2] + opt[n-1][w-weight[n]]

a. What are the steps you have taken to fix the error you identified in the code fragment?

Steps to fix:

- Change option1 = opt[n++][w] to option1 = opt[n-1][w]
- Change profit[n-2] to profit[n] in option2

3. Submit your complete executable code?



```java
public class Knapsack {

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        int W = Integer.parseInt(args[1]);

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

        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];
                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);
            }
        }

        boolean[] take = new boolean[N+1];
        for (int n = N, w = W; n > 0; n--) {
            if (sol[n][w]) { take[n] = true;  w = w - weight[n]; }
            else           { take[n] = false;                    }
        }

        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]);
        }
    }
}
```

magic number:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 2 errors

- Error in while(sum == 0) condition
- Missing semicolon in sum = sum % 10

2. How many breakpoints you need to fix those errors?

Breakpoints: 2 breakpoints needed

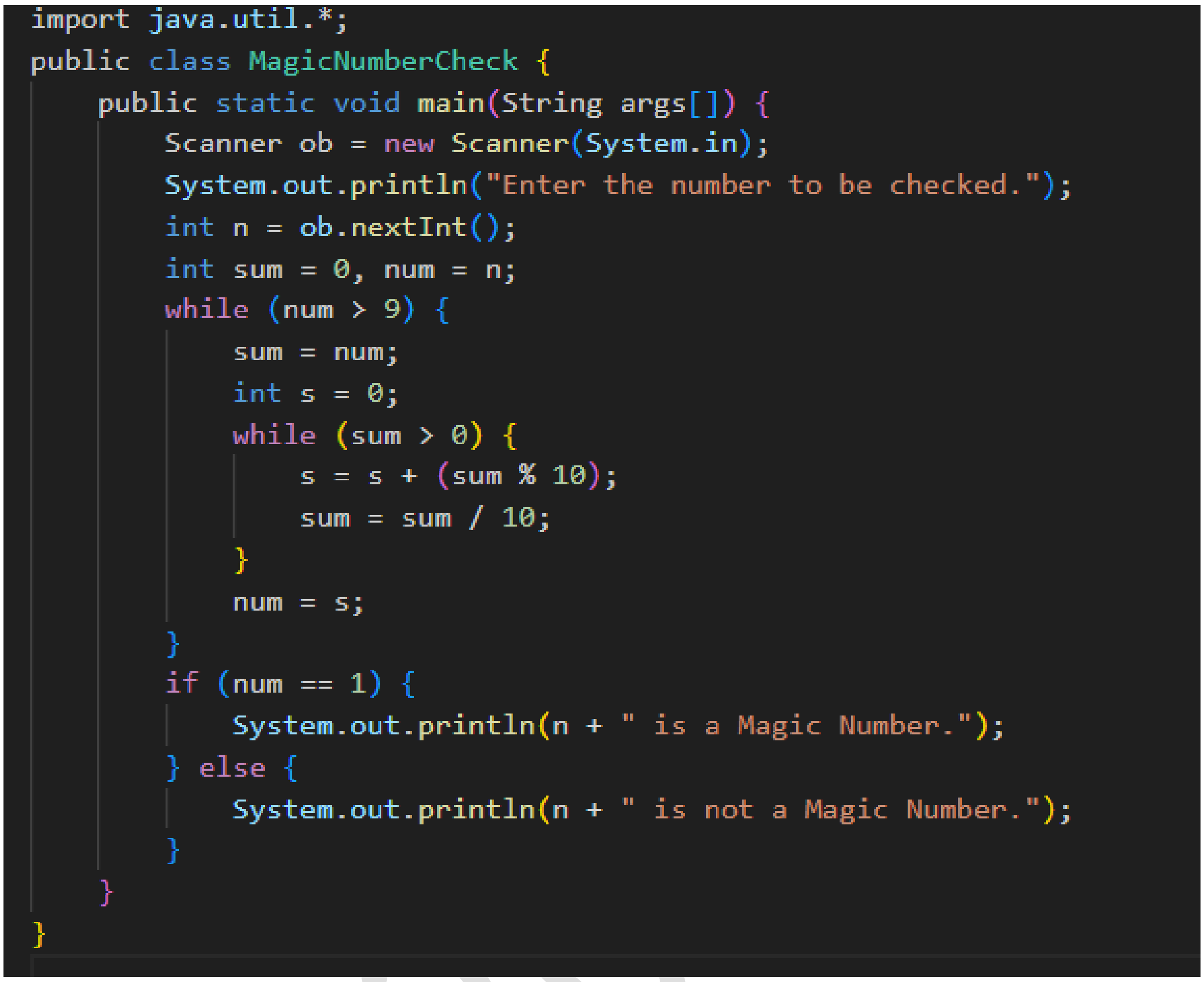
- At while(sum == 0)
- After sum = sum % 10

a. What are the steps you have taken to fix the error you identified in the code fragment?

Steps to fix:

- Change while(sum == 0) to while(sum > 0)
- Add semicolon after sum = sum % 10

3. Submit your complete executable code?



```java
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) {
                s = s + (sum % 10);
                sum = sum / 10;
            }
            num = s;
        }
        if (num == 1) {
            System.out.println(n + " is a Magic Number.");
        } else {
            System.out.println(n + " is not a Magic Number.");
        }
    }
}
```

merge sort:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 3 errors

- Error in int[] left = leftHalf(array+1);
- Error in int[] right = rightHalf(array-1);
- Error in merge(array, left++, right--);

2. How many breakpoints you need to fix those errors?

Breakpoints: 3 breakpoints needed

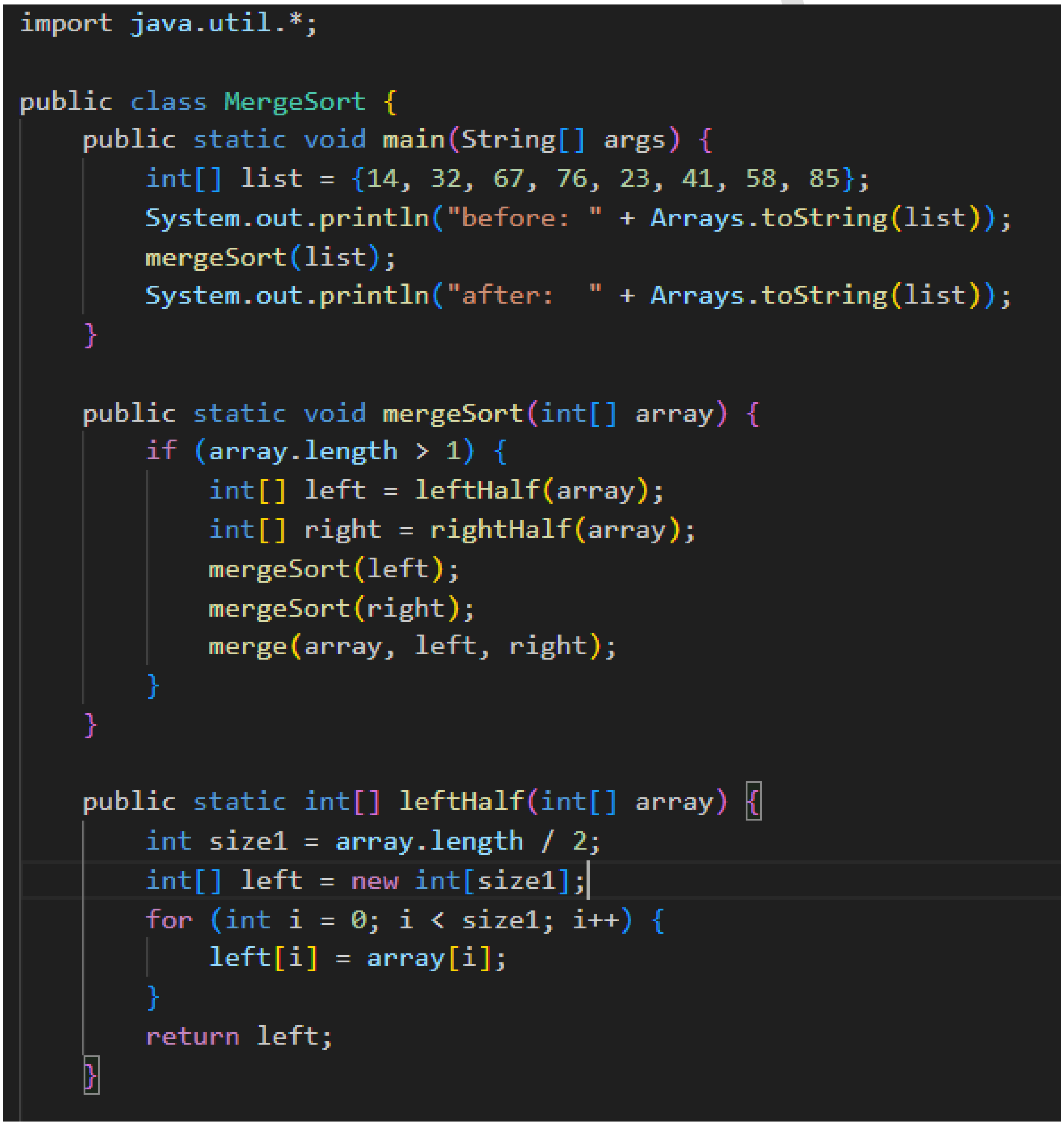
- At int[] left = leftHalf(array+1);
- At int[] right = rightHalf(array-1);
- At merge(array, left++, right--);

a. What are the steps you have taken to fix the error you identified in the code fragment?

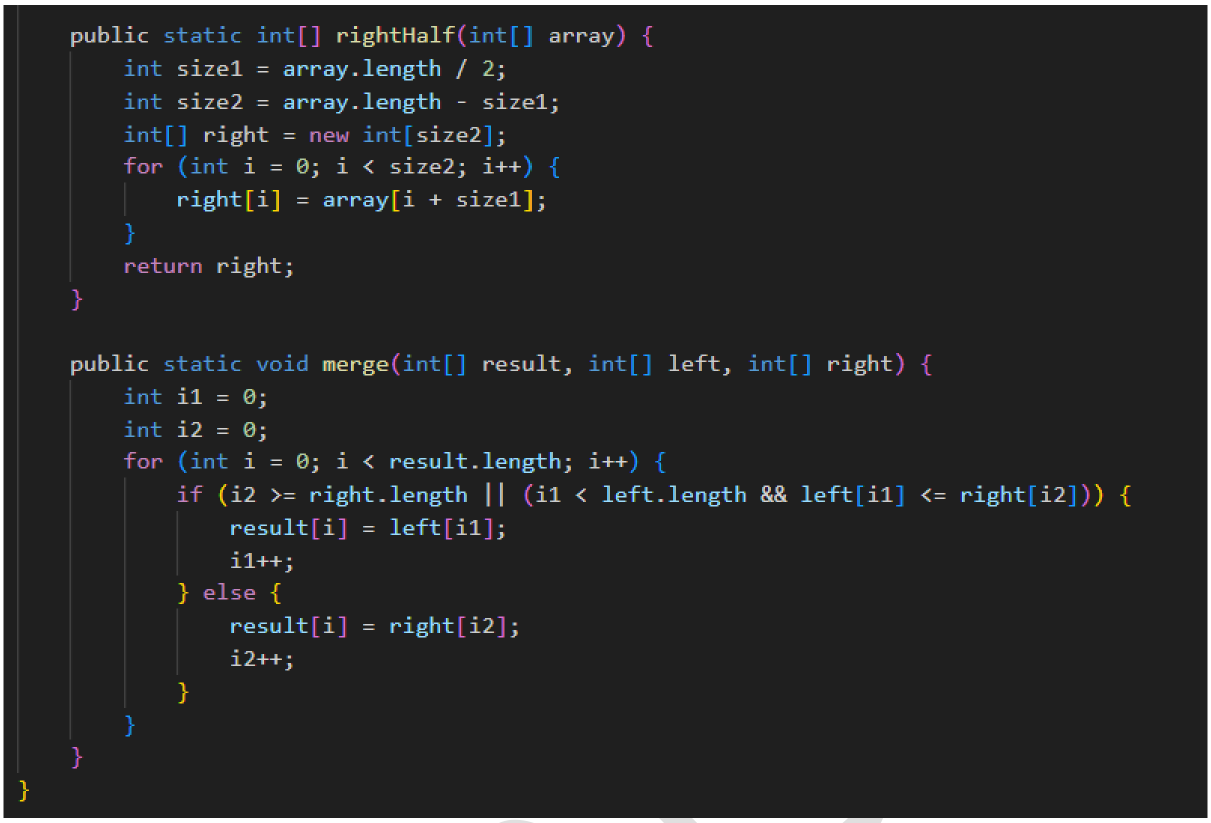
Steps to fix:

- Change array+1 to leftHalf(array)
- Change array-1 to rightHalf(array)
- Change merge(array, left++, right--); to merge(array, left, right);

3. Submit your complete executable code?



```java
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++;
            }
        }
    }
}
```

multiply matrics:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 2 errors

- Error in first[c-1][c-k]
- Error in second[k-1][k-d]

2. How many breakpoints you need to fix those errors?

Breakpoints: 2 breakpoints needed

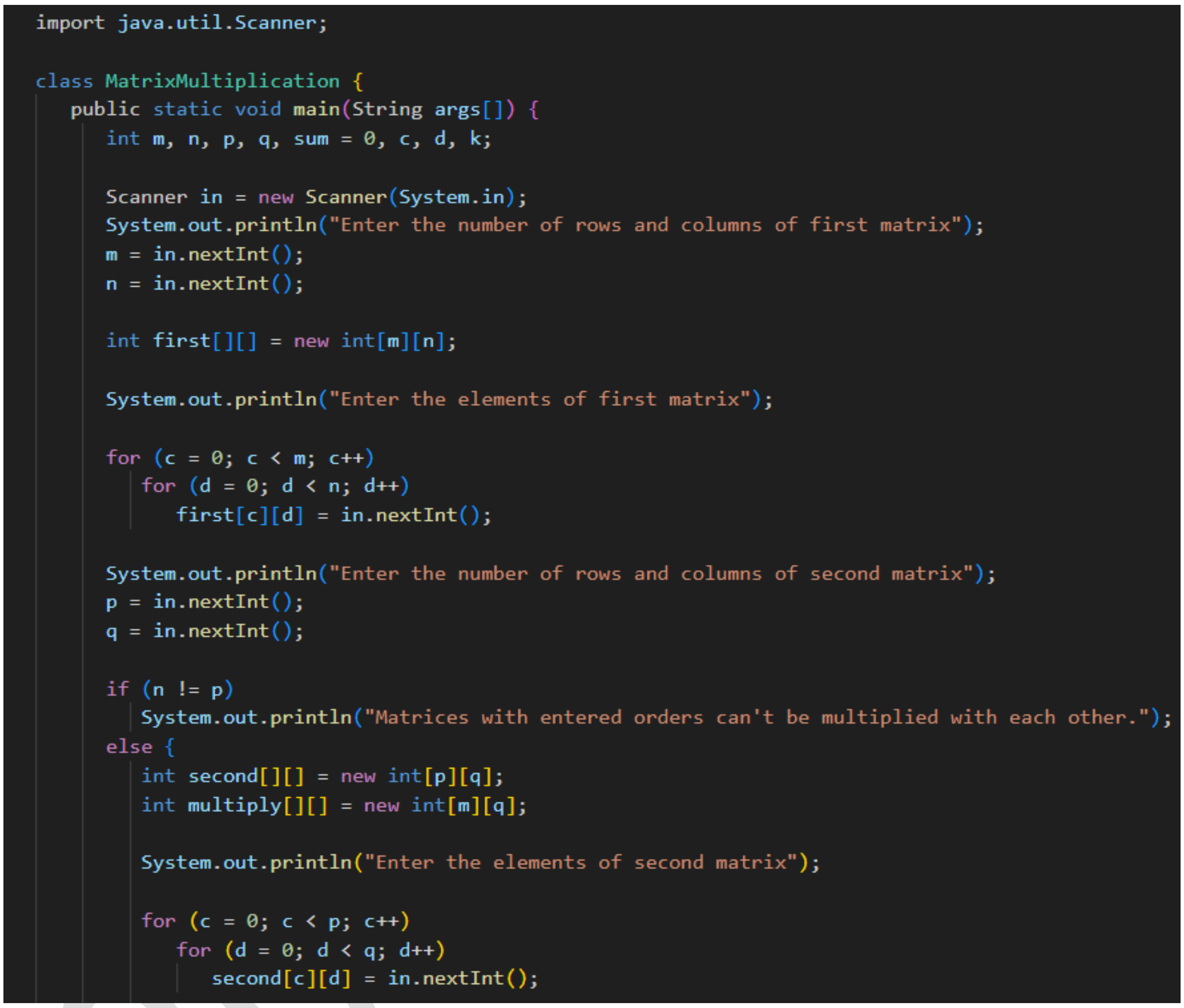
- At first[c-1][c-k]
- At second[k-1][k-d]

a. What are the steps you have taken to fix the error you identified in the code fragment?

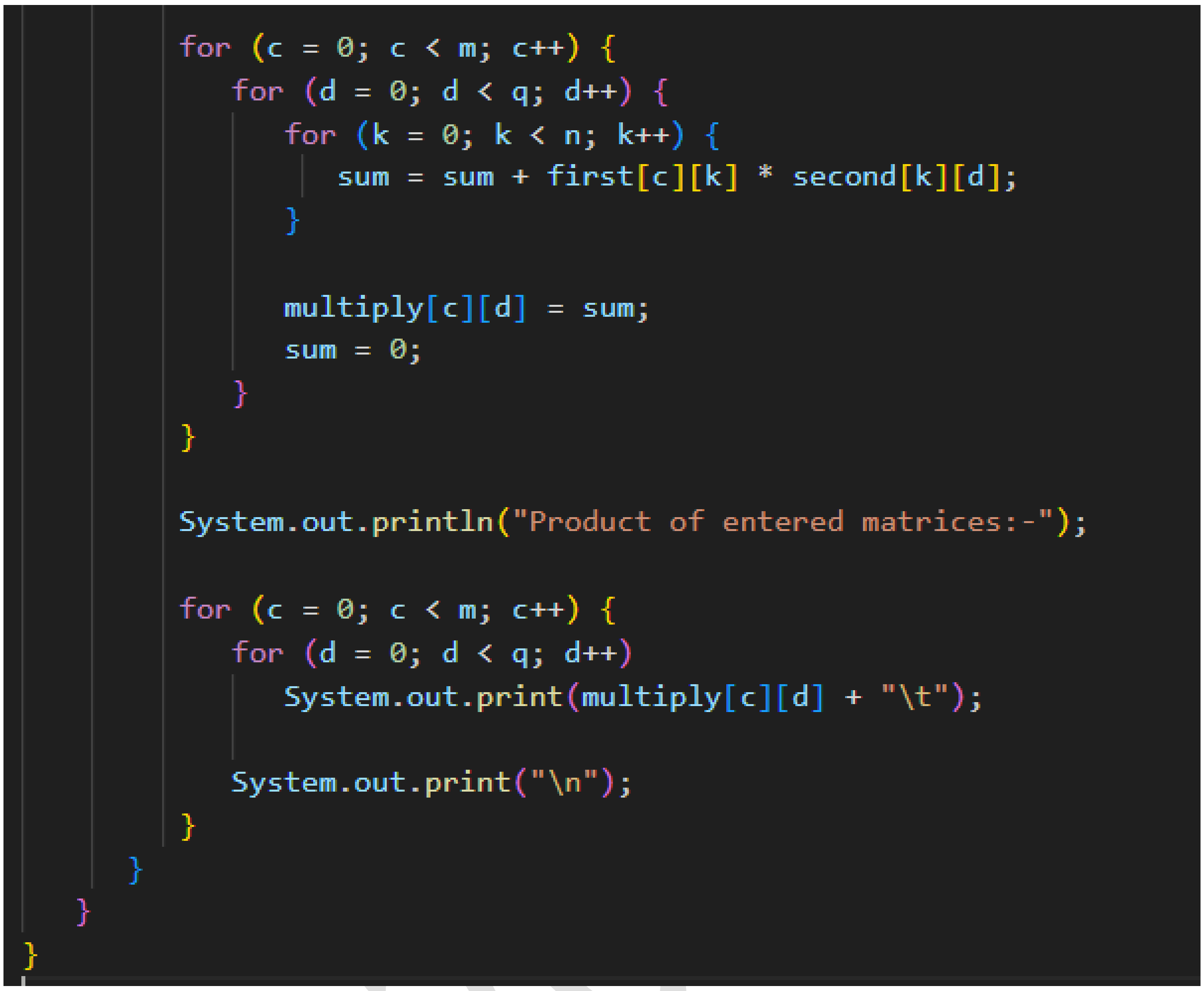
Steps to fix:

- Change first[c-1][c-k] to first[c][k]
- Change second[k-1][k-d] to second[k][d]

3. Submit your complete executable code?



```java
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 first matrix");
      m = in.nextInt();
      n = in.nextInt();

      int first[][] = new int[m][n];

      System.out.println("Enter the elements of 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 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 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 < n; 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");
      }
    }
  }
}
```

Quardatic Probing:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 5 errors

- Error in i + = (i + h / h--) % maxSize; (remove space before +=)
- Error in i = (i + h * h++) % maxSize; (should be i = (i + h * h) % maxSize;)
- Error in while (!key.equals(keys[i])) (change to while (keys[i] != null && !key.equals(keys[i])))
- Error in for (i = (i + h * h++) % maxSize; keys[i] != null; i = (i + h * h++) % maxSize) (change to for (i = (i + h * h) % maxSize; keys[i] != null; i = (i + h * h) % maxSize))
- Error in the comment /** maxSizeake object of QuadraticProbingHashTable **/ (change to /** make object of QuadraticProbingHashTable **/)

2. How many breakpoints you need to fix those errors?

Breakpoints: 5 breakpoints needed

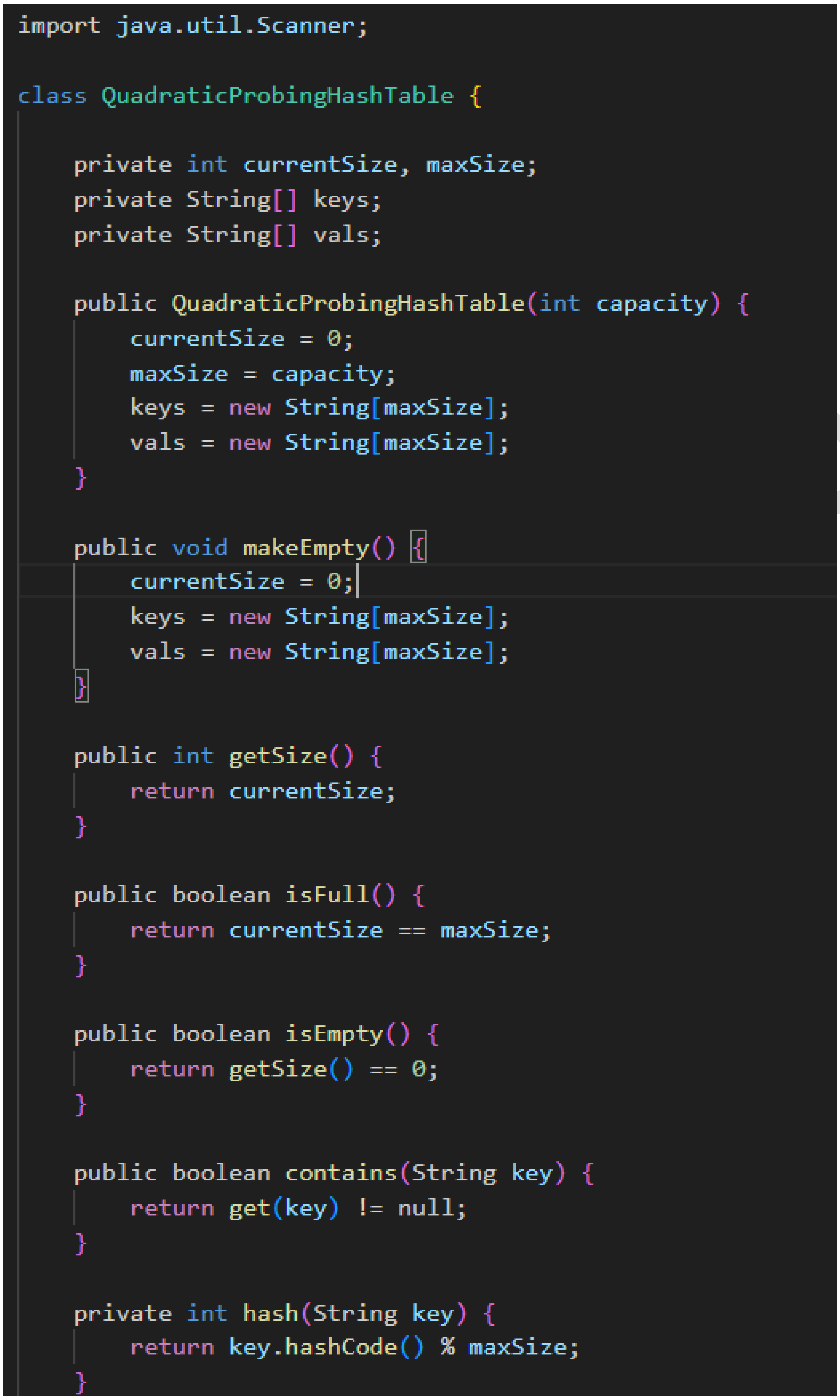
- At i + = (i + h / h--) % maxSize;
- At i = (i + h * h++) % maxSize;
- At while (!key.equals(keys[i]))
- At for (i = (i + h * h++) % maxSize; keys[i] != null; i = (i + h * h++) % maxSize)
- At the comment /** maxSizeake object of QuadraticProbingHashTable **/

a. What are the steps you have taken to fix the error you identified in the code fragment?

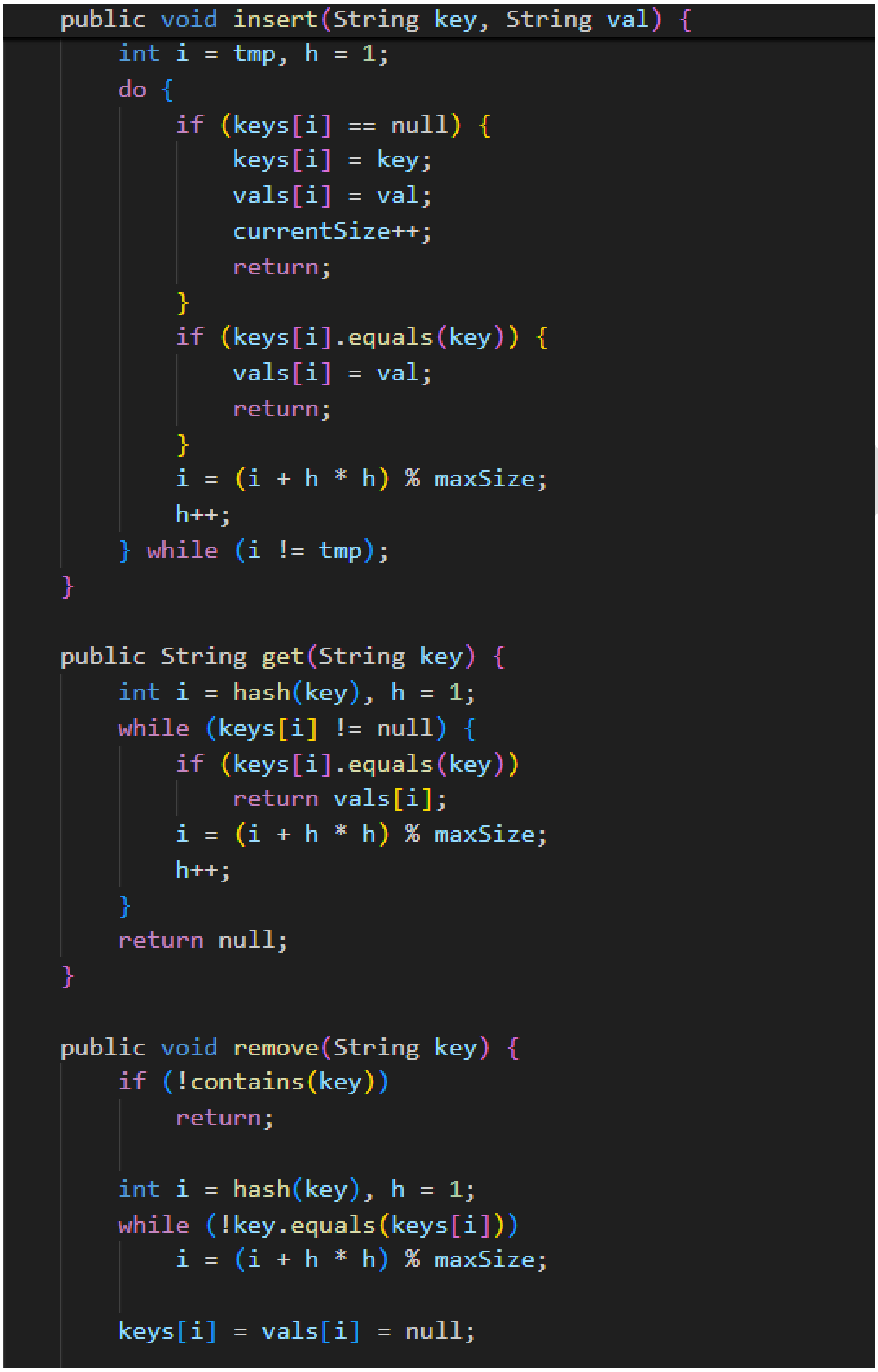
Steps to fix:

- Remove space in i + =
- Change h++ to h in hash calculations
- Update while condition to check for null
- Update for loop conditions
- Correct the comment

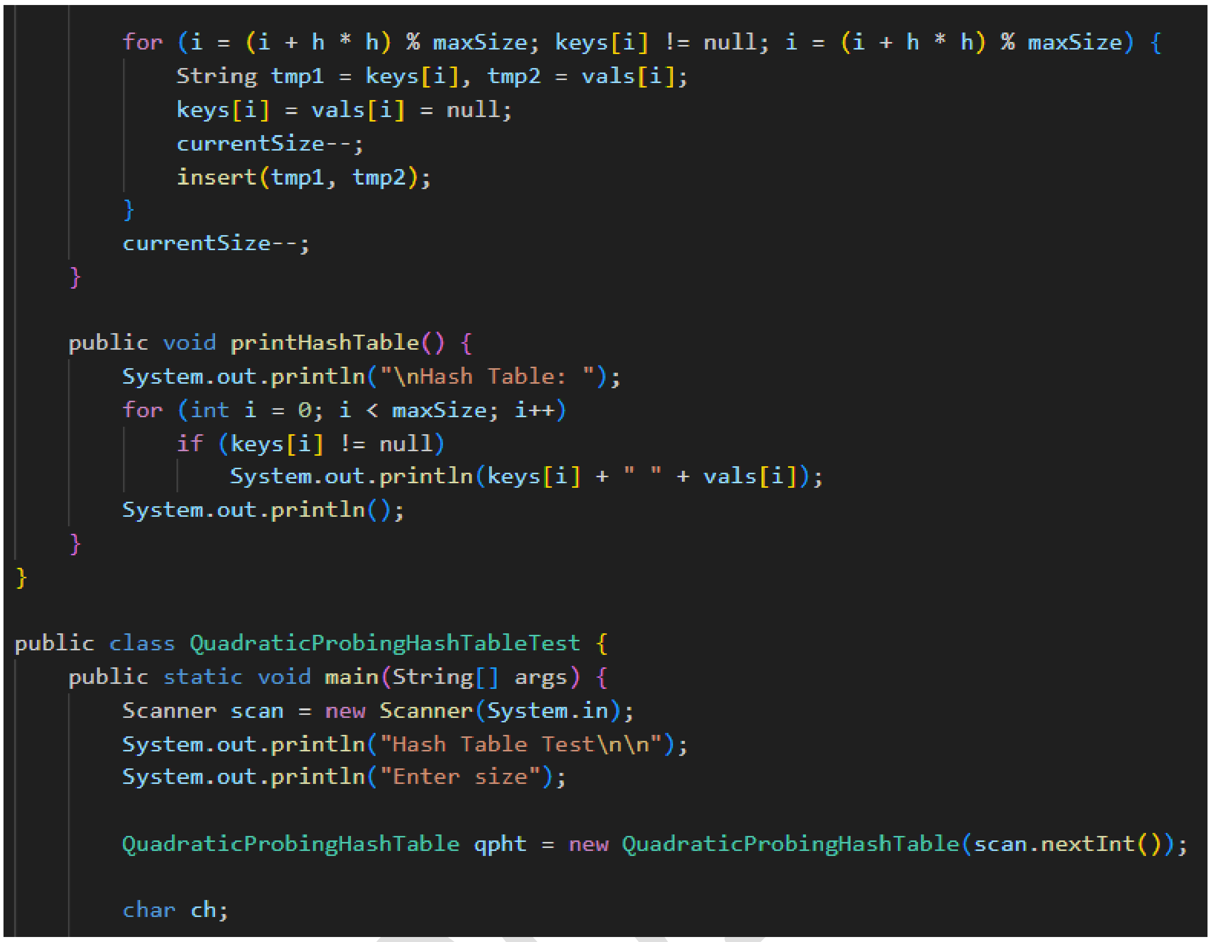
3. Submit your complete executable code?



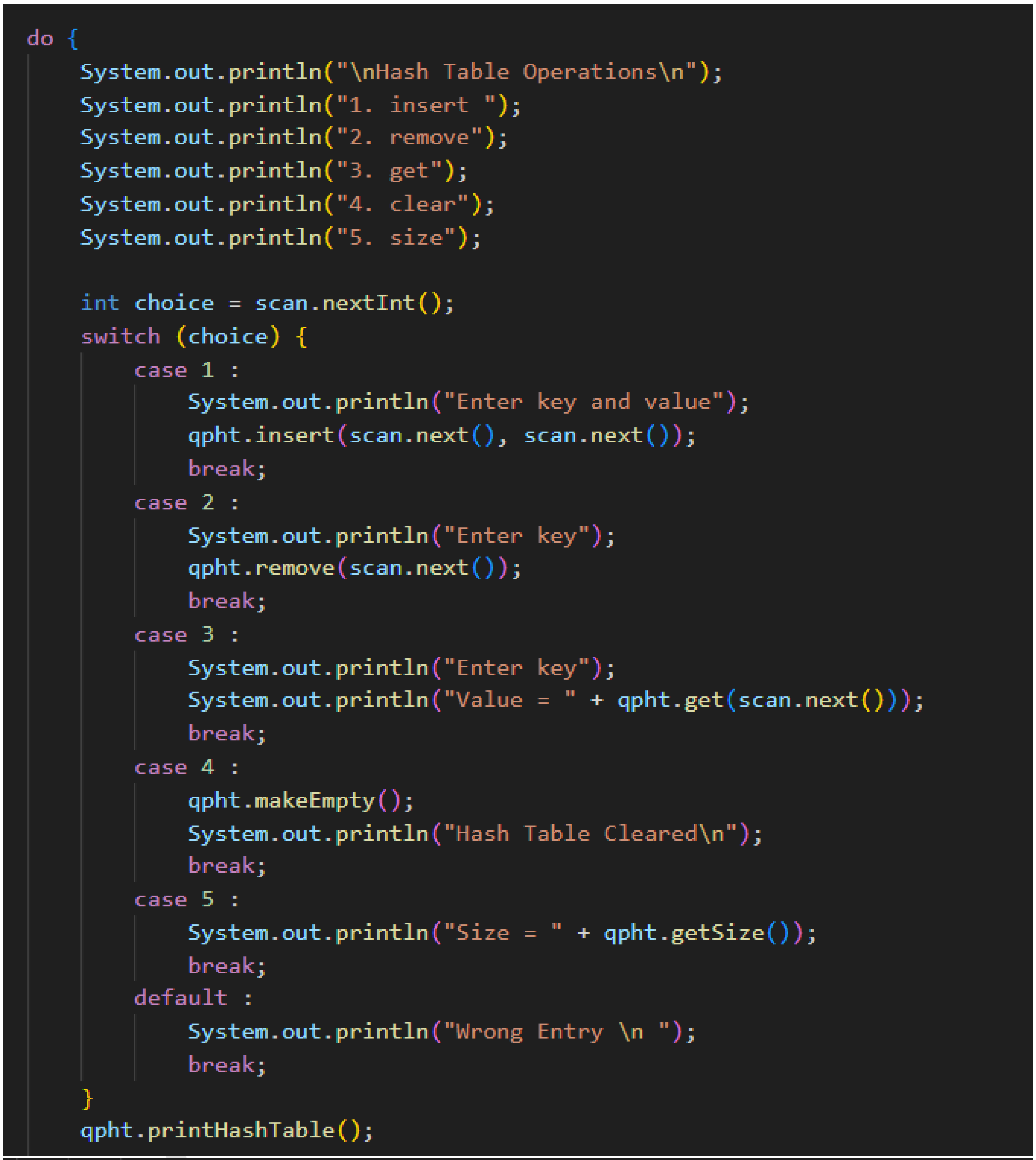
```java
import java.util.Scanner;

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 = new 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;
    }
```



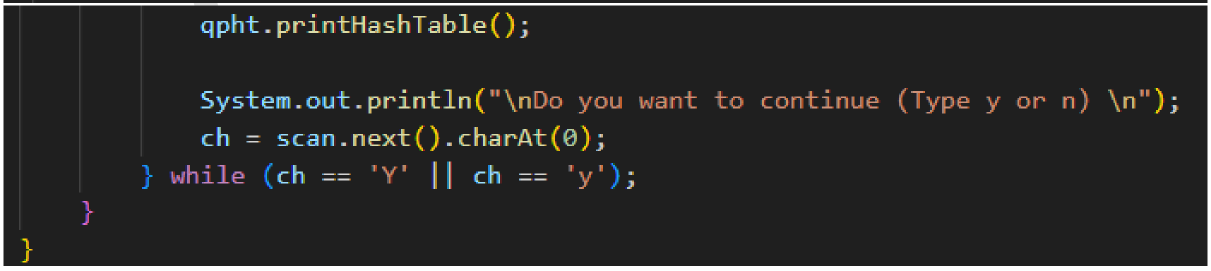
```java
public void insert(String key, String val) {
    int i = tmp, h = 1;
    do {
        if (keys[i] == null) {
            keys[i] = key;
            vals[i] = val;
            currentSize++;
            return;
        }
        if (keys[i].equals(key)) {
            vals[i] = val;
            return;
        }
        i = (i + h * h) % maxSize;
        h++;
    } 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;
        h++;
    }
    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.out.println("Enter size");

        QuadraticProbingHashTable qpht = new QuadraticProbingHashTable(scan.nextInt());

        char ch;
```



```java
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();
```



```java
            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');
    }
}
```

Sorting array:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 5 errors

- Error in class name: public class Ascending _Order (remove space, change to public class AscendingOrder)
- Error in the for loop condition: for (int i = 0; i >= n; i++); (change to for (int i = 0; i < n; i++))
- Error in the swap condition: if (a[i] <= a[j]) (change to if (a[i] > a[j]))
- Error in the final print loop: for (int i = 0; i < n - 1; i++) (change to for (int i = 0; i < n; i++))
- Remove the unnecessary semicolon after the first for loop.

2. How many breakpoints you need to fix those errors?

Breakpoints: 5 breakpoints needed

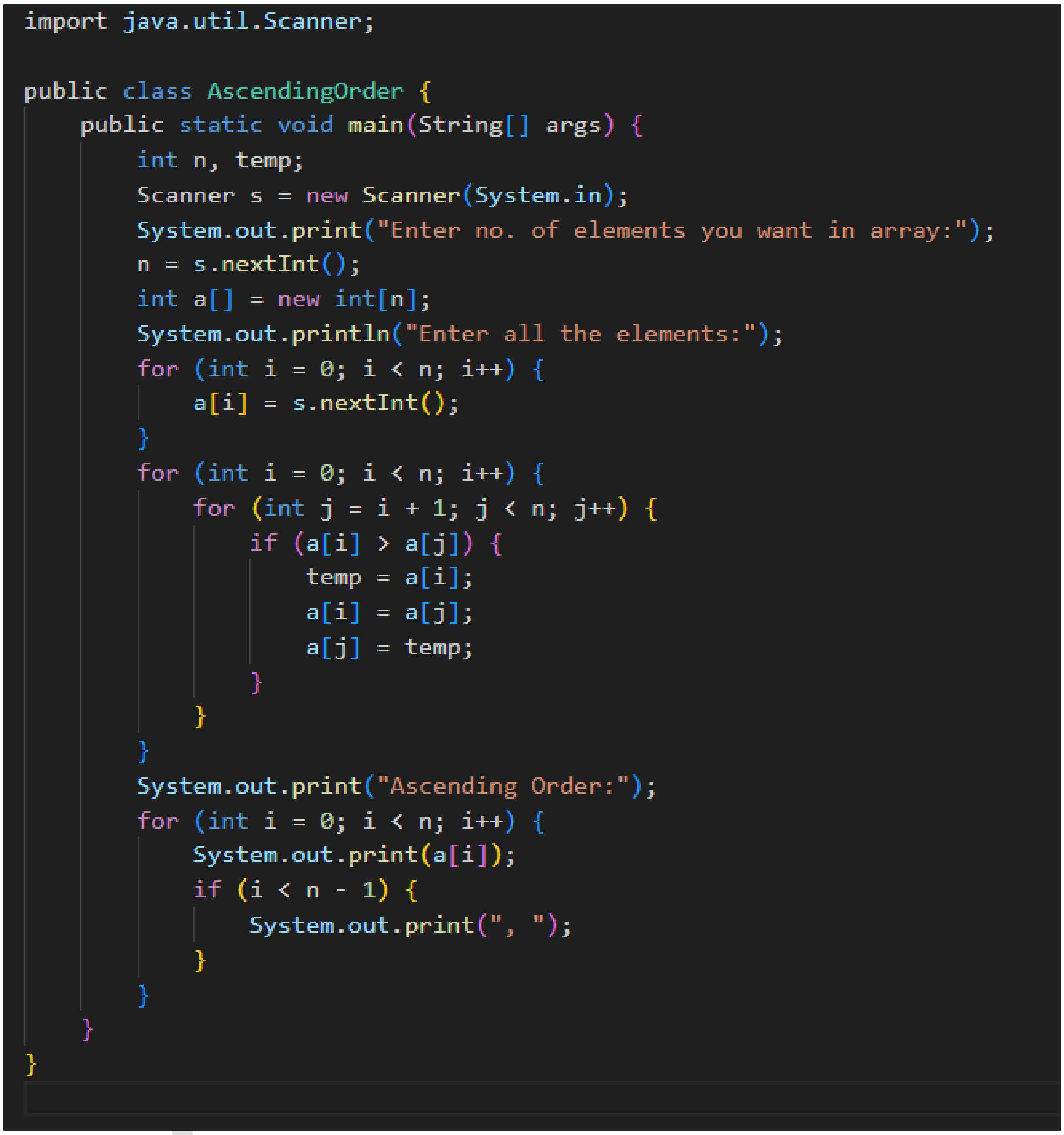
- At public class Ascending _Order
- At for (int i = 0; i >= n; i++);
- At if (a[i] <= a[j])
- At for (int i = 0; i < n - 1; i++)
- At System.out.print(a[n - 1]);

a. What are the steps you have taken to fix the error you identified in the code fragment?

Steps to fix:

- Remove space in class name
- Change loop condition to <
- Change swap condition to >
- Update final print loop to iterate through all elements
- Remove the semicolon after the first for loop

3. Submit your complete executable code?



```java
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 no. of elements you want in 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; i++) {
            System.out.print(a[i]);
            if (i < n - 1) {
                System.out.print(", ");
            }
        }
    }
}
```

Stack Implementation:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 5 errors

- Error in the push method: top--; (should be top++; to push the value onto the stack).
- Error in the push method: stack[top]=value; (this should be after incrementing top).
- Error in the pop method: top++; (should be top--; to remove the value from the stack).
- Error in the display method: for(int i=0;i>top;i++) (should be for(int i=0;i<=top;i++) to iterate correctly).
- In the display method, you need to check for top being -1 before attempting to print elements.

2. How many breakpoints you need to fix those errors?

Breakpoints: 5 breakpoints needed

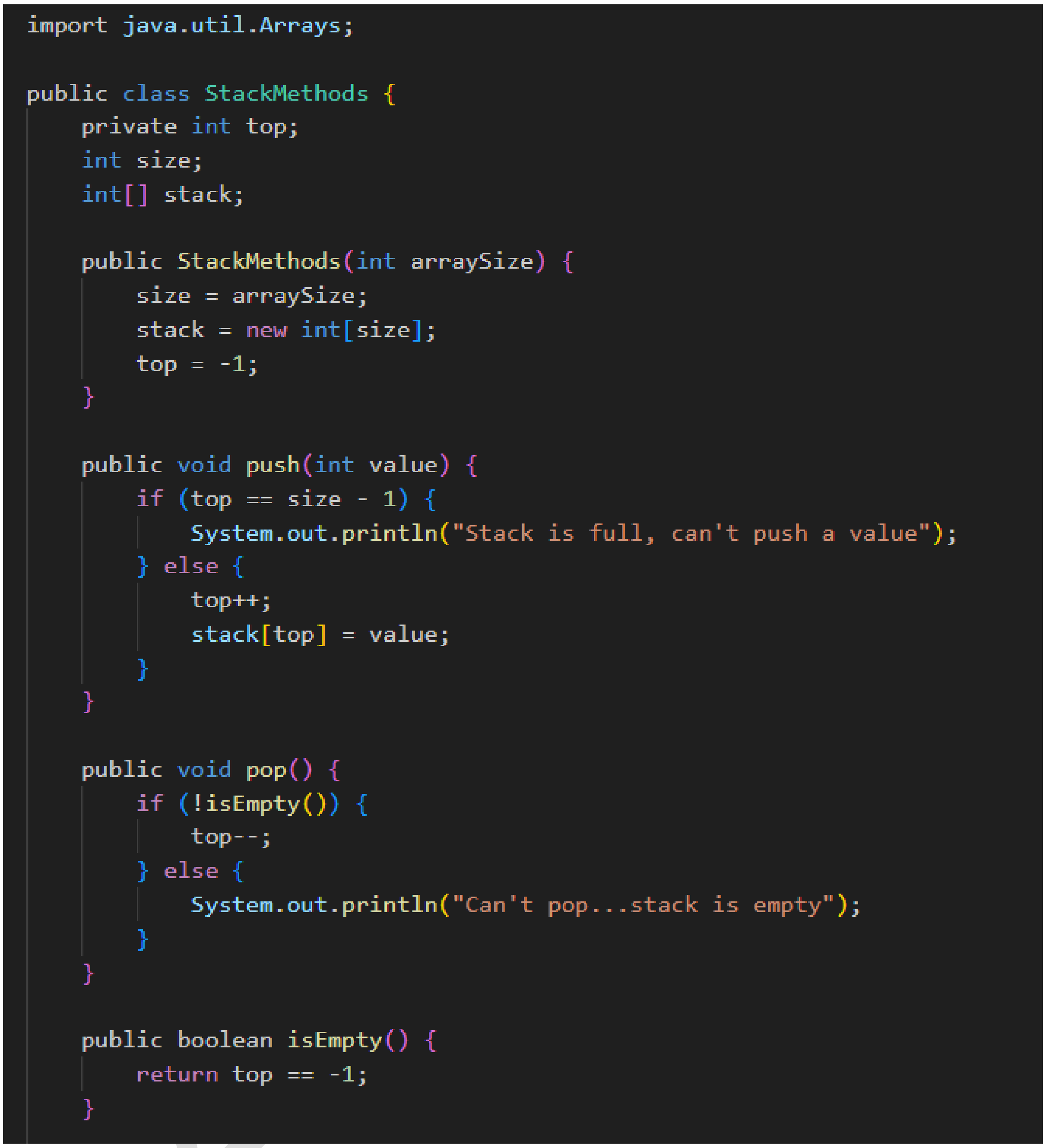
- At top--; in the push method
- At stack[top]=value; in the push method
- At top++; in the pop method
- At for(int i=0;i>top;i++) in the display method
- At System.out.print(stack[i]+ " "); in the display method

a. What are the steps you have taken to fix the error you identified in the code fragment?

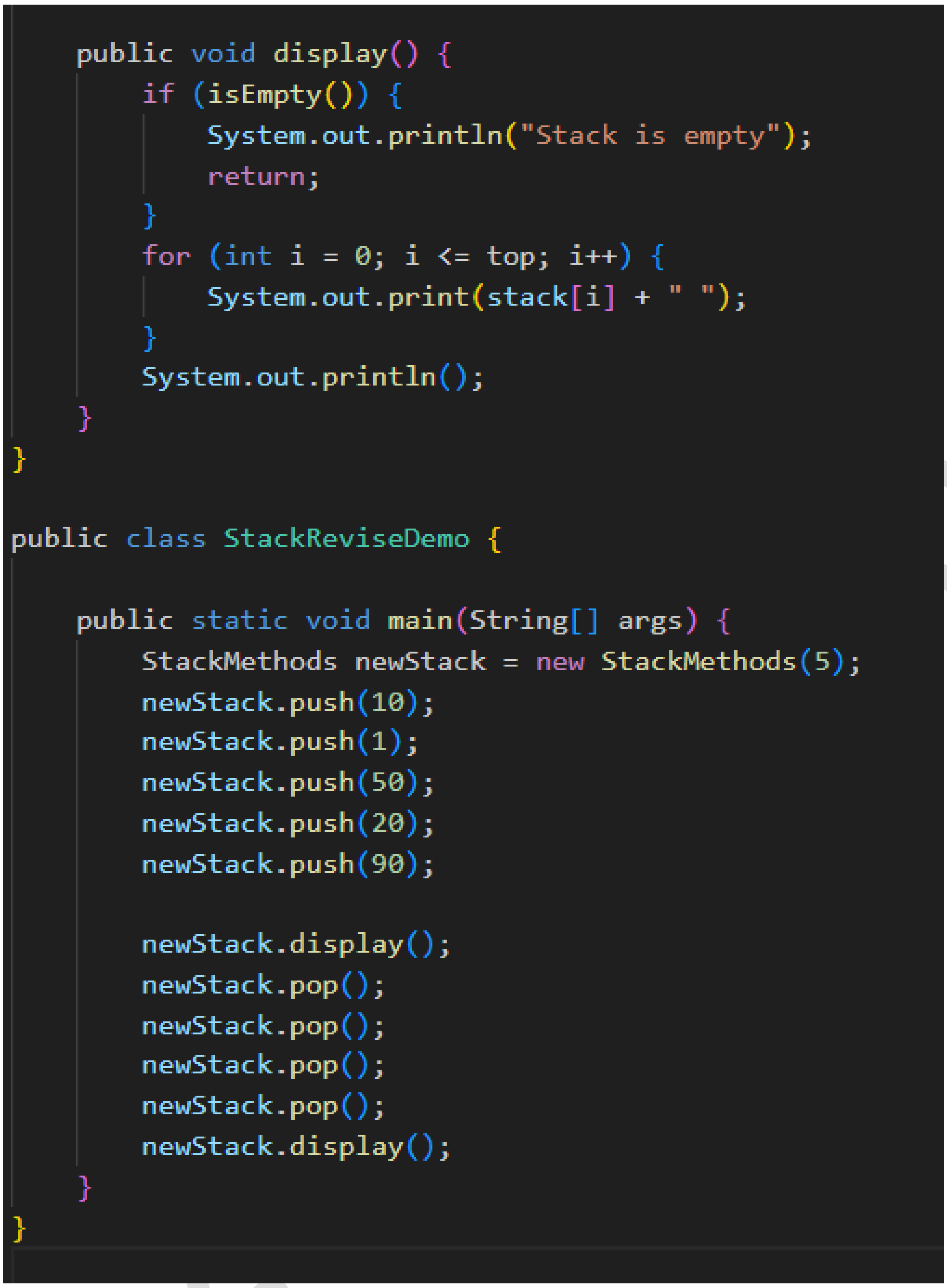
Steps to fix:

- Increment top after checking if the stack is full in push method.
- Update top to decrement when popping in pop method.
- Correct the loop condition in display method to i<=top.
- Add a check for isEmpty() in display method before printing.

3. Submit your complete executable code?



```java
import java.util.Arrays;

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() {
        if (isEmpty()) {
            System.out.println("Stack is empty");
            return;
        }
        for (int i = 0; i <= top; i++) {
            System.out.print(stack[i] + " ");
        }
        System.out.println();
    }
}

public class StackReviseDemo {

    public static void main(String[] args) {
        StackMethods newStack = new StackMethods(5);
        newStack.push(10);
        newStack.push(1);
        newStack.push(50);
        newStack.push(20);
        newStack.push(90);

        newStack.display();
        newStack.pop();
        newStack.pop();
        newStack.pop();
        newStack.pop();
        newStack.display();
    }
}
```

Tower of Hanoi:

1. How many errors are there in the program? Mention the errors you have identified.

Errors: 4 errors

- In the doTowers method, the line doTowers(topN ++, inter--, from+1, to+1) has incorrect syntax. The ++ and -- operators cannot be used directly on method parameters like this.
- The second parameter of the recursive call inter should not be decremented; it should remain the same.
- The third and fourth parameters of the recursive call should be unchanged; they need to be passed as from and to respectively.
- The from + 1 and to + 1 should be inter for the destination parameters in the recursive call.

2. How many breakpoints you need to fix those errors?

Breakpoints: 4 breakpoints needed

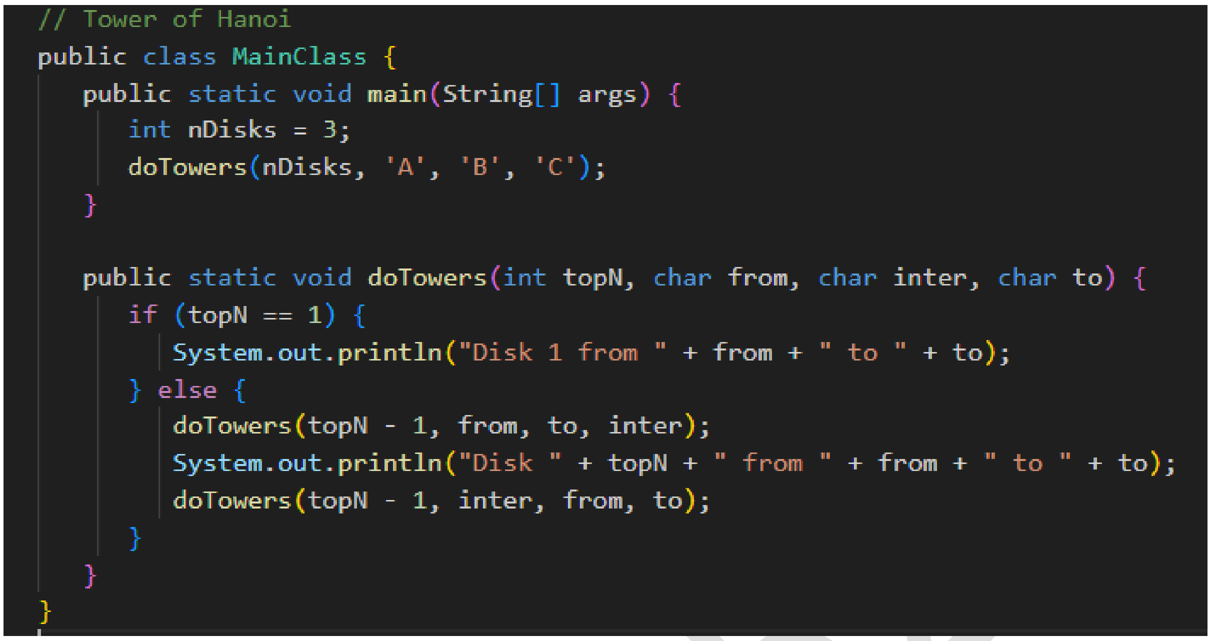
- At the line checking if (topN == 1) to ensure it captures the base case correctly.
- At the line System.out.println("Disk " + topN + " from " + from + " to " + to); to confirm disk movement is correct.
- At the line of the recursive call before the changes.
- At the line doTowers(topN - 1, from, to, inter); to check recursive function flow.

a. What are the steps you have taken to fix the error you identified in the code fragment?

Steps to fix:

- Replace topN ++ and inter-- with topN - 1 and inter respectively in the recursive call.
- Ensure doTowers is called with the correct parameters for the last recursive step.

3. Submit your complete executable code?

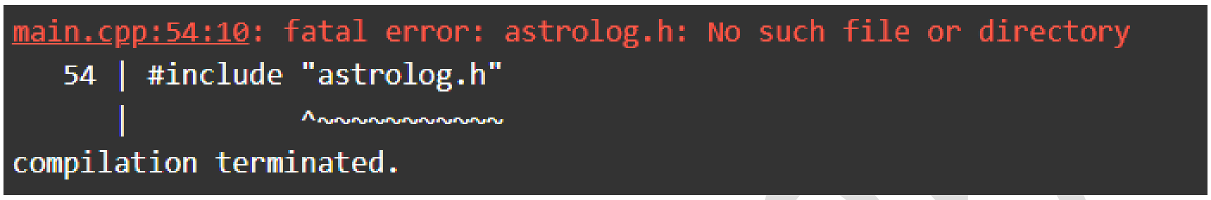


```
// Tower of Hanoi
public class MainClass {
    public static void main(String[] args) {
        int nDisks = 3;
        doTowers(nDisks, 'A', 'B', 'C');
    }

    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);
        }
    }
}
```

Static Analysis Tools:

I used GDB Debugger in that I get this type of error:

1) fatal error (other type of error type in checklist)



```
main.cpp:54:10: fatal error: astrolog.h: No such file or directory
   54 | #include "astrolog.h"
      |          ^~~~~~~~~~~~
compilation terminated.
```