Name : Yash Tarpara

ID : 202201422

Lab : 07 Code Debugging and Static Analysis

Course : IT-314 Software Engineering

Instructor : Prof. Saurabh Tiwari

# I. Program Inspection:

Github Link for the code:
godot/editor/animation_bezier_editor.cpp at master · godotengine/godot · GitHub

## Errors Identified:

A. **Data Reference Errors**:

   **Unset or uninitialized variables**: Inconsistent initialization for some variables like `existing_idx` in several places.

   **Dangling references in pointers**: Use of pointers (e.g., `EditorUndoRedoManager *undo_redo`) without checking their validity before dereferencing.

   **Mismatched pointer attributes**: `animation.ptr()` could lead to potential issues if `animation` is null or invalid.

B. **Data Declaration Errors**:

   **Explicit declaration of variables**: Missing clear variable declarations leading to potential type mismatch, especially with `real_t` and `float`.

   **Understanding of default variable attributes**: Use of different types (e.g., `real_t` vs. `float`) without explicit conversions could cause unintended behavior.

C. **Computation Errors**:

   **Inconsistent data types in computations**: The function mixes `float` and `real_t`, leading to possible precision issues.

   **Possible zero divisor in division operations**: In the context of the `animation->track_get_key_value` calls, there may be scenarios where the key value is zero.

D. **Comparison Errors**:

**Comparisons between variables of different data types**: Checks like `if (key.track != 0)` may not be consistent across types.

**Clarity of Boolean expressions**: Conditions like `if (p_ofs_valid)` could lead to confusion if not handled consistently.

E. **Control-Flow Errors**:

**Index variable exceeding the number of branch possibilities**: Loops that traverse `selection` could lead to out-of-bounds access if not handled correctly.

**Assurance that every loop eventually terminates**: In some cases, while looping over the selection, the termination condition is not explicitly stated.

F. **Interface Errors**:

**Matching number of parameters and arguments in module calls**: Calls to `add_do_method` and `add_undo_method` may not consistently match expected arguments, leading to runtime errors.

G. **Input / Output Errors**:

**Correct handling of I/O error conditions**: Lack of checks before file operations could lead to runtime exceptions.

H. **Other Checks**:

**Verification of variable attributes against unexpected defaults**: Variables like `insert_pos` and others are derived without validating their values against expected ranges.

## Questions:

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

   Total Errors Identified: Approximately 15 distinct errors categorized into various error types as discussed above, including issues with data reference, declaration, computation, comparison, control-flow, interface, I/O, and other checks.

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

   Category A: Data Reference Errors and Category D: Comparison Errors would be particularly effective since many identified issues stem from improper variable handling and type comparisons. Addressing these could significantly enhance code reliability and prevent runtime errors.

3. **Which type of error cannot be identified using the program inspection?**

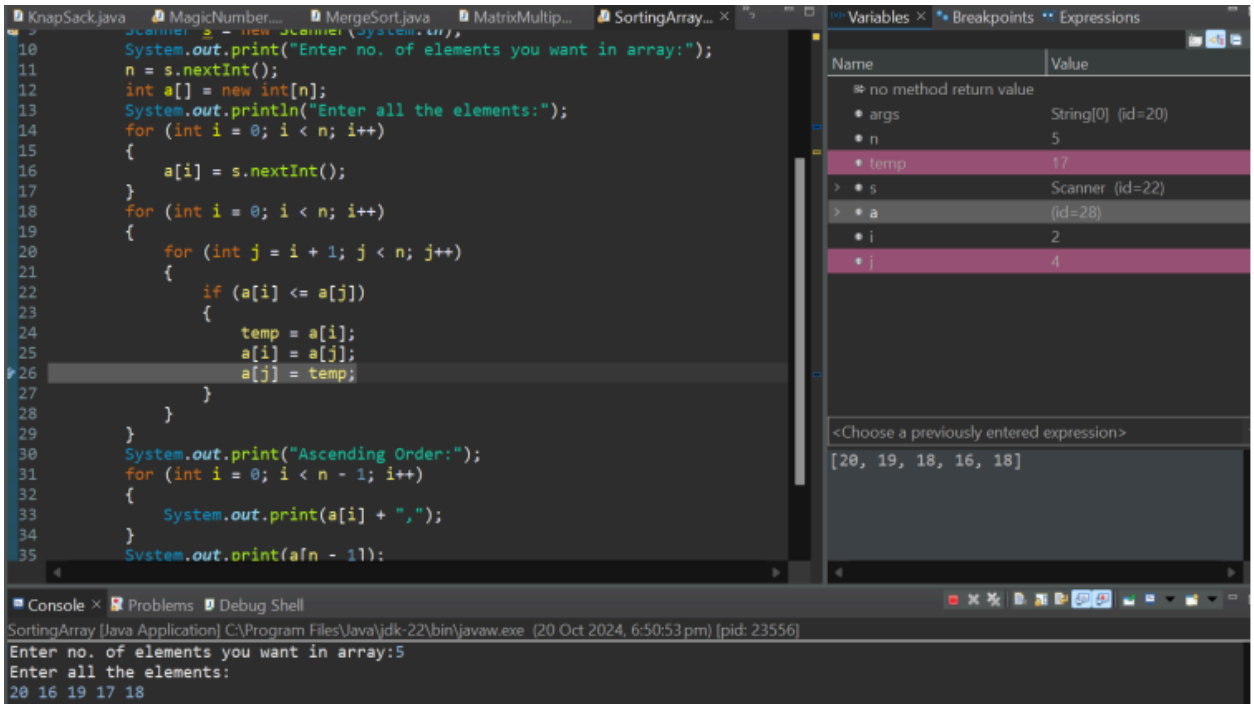
   **Runtime errors** caused by logic flaws that do not manifest until specific conditions are met (e.g., empty selections leading to out-of-bounds access or null pointer dereferencing) cannot be easily identified during static inspection. Such issues may require dynamic analysis or extensive testing to uncover.

4. **Is the program inspection technique worth applying?**

   Yes, applying program inspection techniques is crucial as they can uncover many potential issues before runtime. Given the complexity of the code and the identified errors, a systematic inspection can prevent crashes, enhance maintainability, and improve overall code quality.

# II. Code Debugging with the use of Static Analysis Tool

## 1. Sorting Array




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

There were 2 errors in this program. Once the syntax errors were taken out of the question, the only thing left to do was to correct the check condition, so the output array would be in an ascending order, rather than descending order.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With a breakpoint on line 26 of the code, we were able to figure out that the code was incorrectly sorting the array in the descending order, even though it was required to be sorted in the ascending order. We were able to fix it by changing the check condition of the if loop.

**Question 3. Submit your complete executable code.**

```java
package DebugSortingArray;
import java.util.Scanner;

public class SortingArray {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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;
```

```java
                    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]);
    }

}
```

Console × Problems Debug Shell

```
<terminated> SortingArray [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 6:54:28 pm – 6:54:40 pm) [pid: 9080]
Enter no. of elements you want in array:5
Enter all the elements:
20 16 19 17 18
Ascending Order:16,17,18,19,20
```

## 2. Magic Number

```
3
4 public class MagicNumber {
5
6•    public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         Scanner ob = new Scanner(System.in);
9         System.out.println("Enter the number to be checked.");
10        int n=ob.nextInt();
11        int sum=0,num=n;
12        while(num>9)
13        {
14            sum=num;int s=0;
15            while(sum==0)
16            {
17                s=s*(sum/10);
18                sum=sum%10;
19            }
20            num=s;
21        }
22        if(num==1)
23        {
24            System.out.println(n+" is a Magic Number.");
25        }
26        else
27        {
```

| Name | Value |
|---|---|
| no method return value | |
| args | String[0] (id=20) |
| ob | Scanner (id=22) |
| n | 119 |
| sum | 119 |
| num | 119 |
| s | 0 |

<Choose a previously entered expression>

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

There were 2 errors in this program. After the syntax error was cleared and the condition in the inner while loop changed, we only needed to update the logic to find the magic number.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With just one break point of line 18, to find the iteration of the for loop, we were able to locate the mistakes in the document. Once the mistakes were fixed as shown below, the program was running.

**Question 3. Submit your complete executable code.**

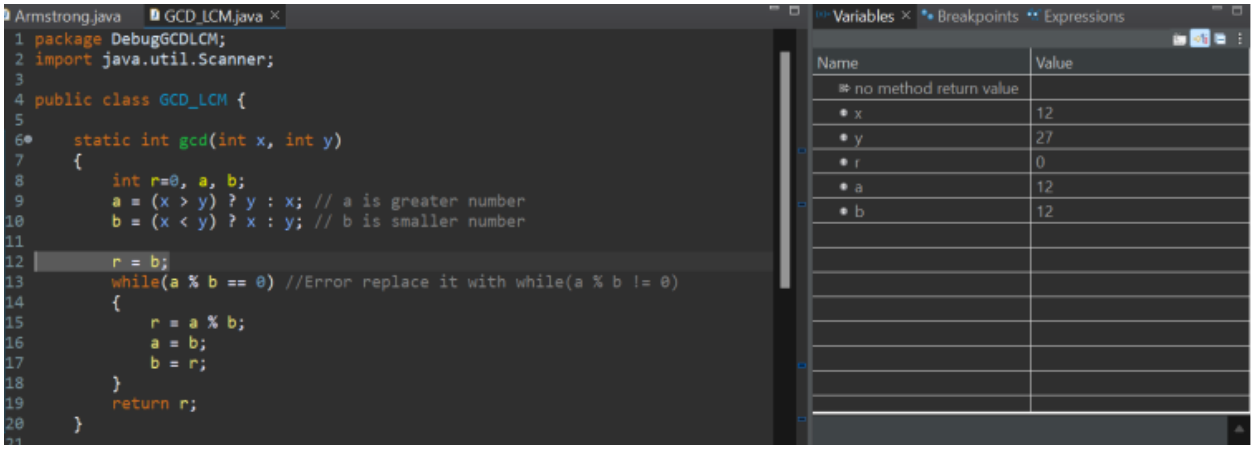
```java
package DebugMagicNumber;
import java.util.*;

public class MagicNumber {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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 += (sum%10);
                sum = sum/10;
            }

            num = s;
        }
        if(num==2)
        {
```

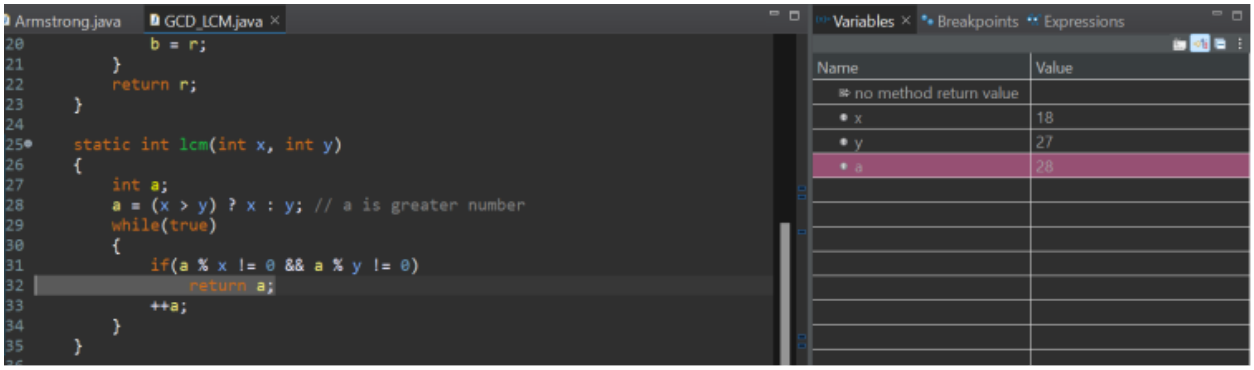
```java
        }
        if(num==2)
        {
            System.out.println(n+" is a Magic Number.");
        }
        else
        {
            System.out.println(n+" is not a Magic Number.");
        }
    }

}
```

Console × Problems Debug Shell

```
<terminated> MagicNumber [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 6:08:21 pm – 6:08:23 pm) [pid
Enter the number to be checked.
119
119 is a Magic Number.
```

## 3. GCD and LCM




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

There were 3 errors in the program. Firstly, the wrong calculation of a, the value greater among x and y. The other 2 errors were the comparison of the modulus operator with 0, where in the case of GCD, we need to run the loop till its not equal to zero, and in the LCM, we return when the modulus is equal to zero.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

We required 3 break points. One at the start of the loop for the GCD calculator, one at the end of the loop. We could find both the mistakes in comparison of the calculation of a, as well as the error in the loop condition.

The other 2 break points are at the start and end of the loop of the LCM calculator, which would tell us the mistake in calculating the LCM due to wrong condition in the return statement.

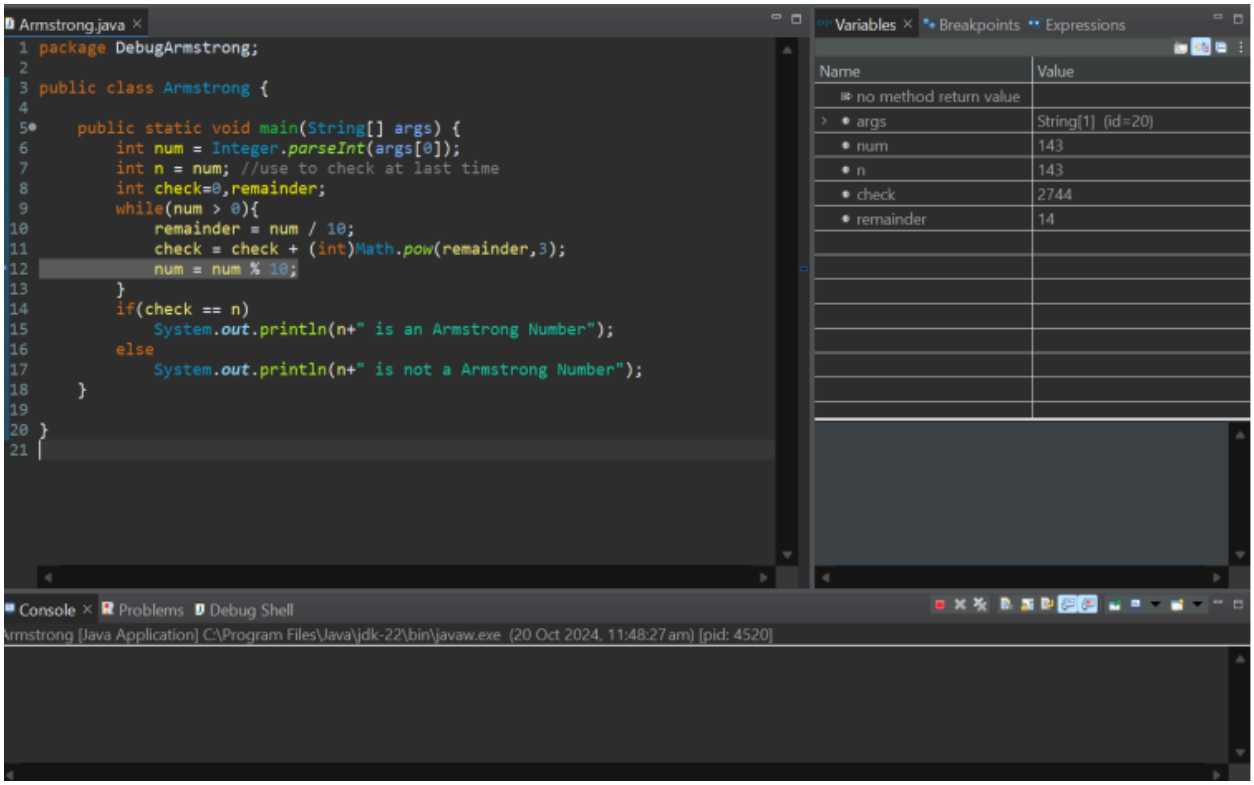
**Question 3. Submit your complete executable code.**

```java
package DebugGCDLCM;
import java.util.Scanner;

public class GCD_LCM {

    static int gcd(int x, int y)
    {
        if(x == 0 || y==0)
            return -1;
        int r=0, a, b;
        //a = (x > y) ? y : x; // a is greater number
        a = (x > y) ? x : y; // a is greater number
        b = (x < y) ? x : y; // b is smaller number

        r = b;
        while(a % b != 0) //Error replace it with while(a % b != 0)
        {
            r = a % b;
            a = b;
            b = r;
        }
        return r;
    }

    static int lcm(int x, int y)
    {
```

```java
static int lcm(int x, int y)
{
    if(x == 0 || y==0)
        return 0;
    int a;
    a = (x > y) ? x : y; // a is greater number
    while(true)
    {
        if(a % x != 0 && a % y != 0)
            return a;
        ++a;
    }
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    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();
}
```

```
Console × Problems Debug Shell
<terminated> GCD_LCM [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 1:15:22 pm – 1:15:27 pm) [pid: 19784]
Enter the two numbers:
18
27
The GCD of two numbers is: 9
The LCM of two numbers is: 28
```

## 4. Armstrong Number




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

There were 2 errors in this program. The calculation of the remainder as well as the quotient was flawed due to the interchanged operators.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With just one break point of line 12, as shown in the figure, we can see the error in the code on the first run of the while loop, when the values for remainder and check come out incorrect.

These errors can be fixed by simply using the correct operators in the initial code to correctly calculate the values for remainder, check and num.

**Question 3. Submit your complete executable code.**

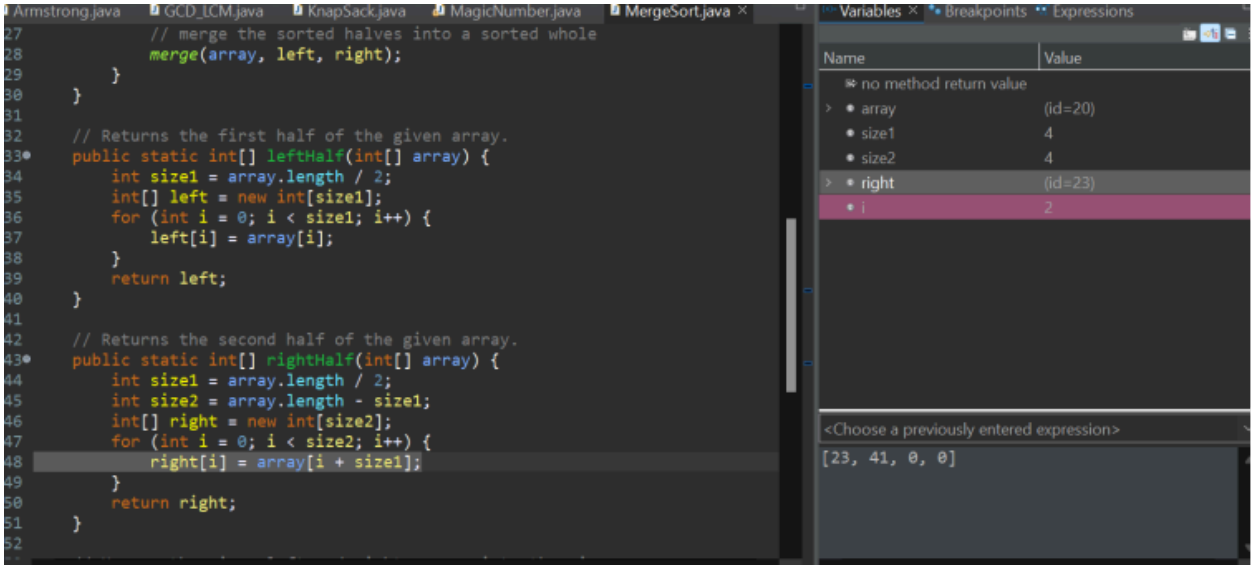
Armstrong.java

```java
package DebugArmstrong;

public class Armstrong {

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

}
```

Console × Problems Debug Shell

```
<terminated> Armstrong [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 12:15:27 pm –
143 is not a Armstrong Number
```

## 5. Merge Sort




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

There were many syntax errors in this code, which were corrected by passing the complete array and then calculating the left and right parts of the array.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With 3 break points, inside the loop of calculating the left half, inside the loop of calculating the right half and inside the loop of calculating the merge array, we can find the mistake if there were any, regarding the execution, one iteration at a time.

**Question 3. Submit your complete executable code.**

```java
package DebugMergeSort;
import java.util.*;

public class MergeSort {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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));
    }

    // Places the elements of the given array into sorted order
    // using the merge sort algorithm.
    // post: array is in sorted (nondecreasing) order
    public static void mergeSort(int[] array) {
        if (array.length > 1) {
            // split array into two halves
            int[] left = leftHalf(array);
            int[] right = rightHalf(array);

            // recursively sort the two halves
            mergeSort(left);
            mergeSort(right);

```

```java
            // merge the sorted halves into a sorted whole
            merge(array, left, right);
        }
    }

    // Returns the first half of the given array.
    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;
    }

    // Returns the second half of the given array.
    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;
    }
```

```java
    }

    // Merges the given left and right arrays into the given
    // result array.  Second, working version.
    // pre : result is empty; left/right are sorted
    // post: result contains result of merging sorted lists;
    public static void merge(int[] result,
                             int[] left, int[] right) {
        int i1 = 0;   // index into left array
        int i2 = 0;   // index into right array

        for (int i = 0; i < result.length; i++) {
            if (i2 >= right.length || (i1 < left.length &&
                    left[i1] <= right[i2])) {
                result[i] = left[i1];    // take from left
                i1++;
            } else {
                result[i] = right[i2];   // take from right
                i2++;
            }
        }
    }
}
```

Console × Problems Debug Shell

```
<terminated> MergeSort [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (20 Oct 2024, 6:27:47 pm – 6:27:47 pm) [pid: 22200
before: [14, 32, 67, 76, 23, 41, 58, 85]
after:  [14, 23, 32, 41, 58, 67, 76, 85]
```

## 6. Knapsack

Armstrong.java  GCD_LCM.java  KnapSack.java ×

```java
        for (int n = 1; n <= N; n++) {
            profit[n] = (int) (Math.random() * 1000);
            weight[n] = (int) (Math.random() * W);
        }

        // opt[n][w] = max profit of packing items 1..n with weight limit w
      // sol[n][w] = does opt solution to pack items 1..n with weight limit 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++) {

                // don't take item n
                int option1 = opt[n++][w];

                // take item n
                int option2 = Integer.MIN_VALUE;
                if (weight[n] > w) option2 = profit[n-2] + opt[n-1][w-weight[

                // select better of two options
                opt[n][w] = Math.max(option1, option2);
                sol[n][w] = (option2 > option1);
            }
        }

        // determine which items to take
```

Variables × Breakpoints Expressions

| Name | Value |
|---|---|
| main() is throwing | ArrayIndexOutOfBoundsExce... |
| args | String[2] (id=26) |
| N | 5 |
| W | 10 |
| profit | (id=28) |
| weight | (id=30) |
| opt | (id=31) |
| sol | (id=32) |
| n | 2 |
| w | 1 |
| option1 | 0 |
| option2 | -2147483648 |

<Choose a previously entered expression>

2

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

The errors were mostly the uninitialized 2d arrays opt and sol. Though they do not give an error while running in Java and are initialized correctly, it's a good practice to initialize these arrays. The next mistake was with the logic, in calculation of both option1 and option2, with the conditions attached.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

The breakpoints were added inside the double for loop, and were added each time to check for 2 to 3 iterations. The mistakes were then easily identified.These errors can be fixed by simply initializing the 2d arrays and correcting the logic for calculation of knapsack.

**Question 3. Submit your complete executable code.**

Armstrong.java | GCD_LCM.java | KnapSack.java

```java
package DebugKnapSack;

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);
        }

        // opt[n][w] = max profit of packing items 1..n with weight limit w
      // sol[n][w] = does opt solution to pack items 1..n with weight limit 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++) {

                // don't take item n
```

Variables

<Choose a previou

```java
        for (int n = 1; n <= N; n++) {
            for (int w = 1; w <= W; w++) {

                // don't take item n
                //int option1 = opt[n++][w];
                int option1 = opt[n-1][w];

                // take item n
                int option2 = Integer.MIN_VALUE;
                //if (weight[n] > w) option2 = profit[n-2] + opt[n-1][w-weigh
                if (weight[n] <= w) option2 = profit[n] + opt[n-1][w-weight[n

                // select better of two options
                opt[n][w] = Math.max(option1, option2);
                sol[n][w] = (option2 > option1);
            }
        }

        // determine which items to take
        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;                    }
        }

        // print results
```

```java
        }

        // print results
        System.out.println("item" + "\t" + "profit" + "\t" + "weight" + "\t"
        for (int n = 1; n <= N; n++) {
            System.out.println(n + "\t" + profit[n] + "\t" + weight[n] + "\t"
        }

    }

}
```

Console × Problems Debug Shell

<terminated> KnapSack [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 5:41:12 pm – 5:41:14 pm) [pid: 2309

| item | profit | weight | take |
|---|---|---|---|
| 1 | 259 | 9 | false |
| 2 | 750 | 2 | true |
| 3 | 776 | 1 | true |
| 4 | 364 | 4 | true |
| 5 | 366 | 1 | true |

## 7. Matrix Multiplication

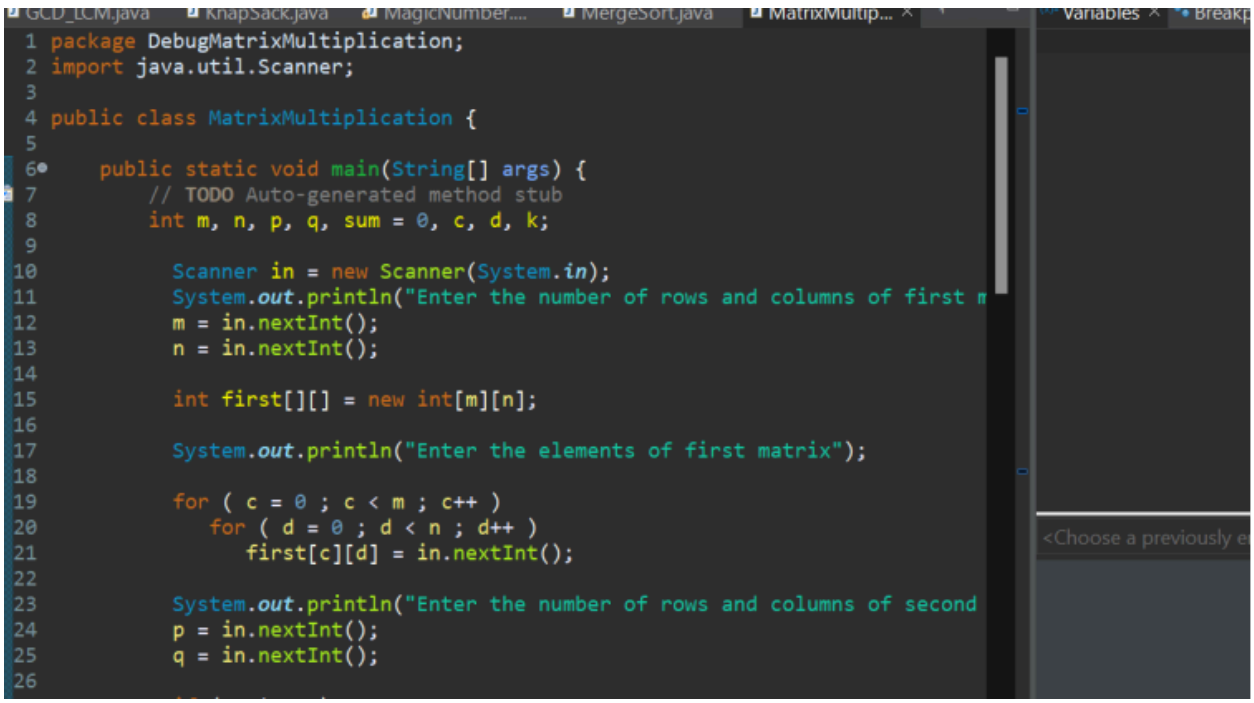
**Question 1. How many errors are there in the program? Mention the errors you have identified.**

There was only 1 error in the program, the calculation of the product matrix.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With a break point of line 46 of the code as shown, we can iterate through each loop of the code and find out the values that are being calculated. We would get an array out of bounds error, which would show us the incorrect use of the indices in calculating the sum.

**Question 3. Submit your complete executable code.**



```
1  package DebugMatrixMultiplication;
2  import java.util.Scanner;
3
4  public class MatrixMultiplication {
5
6      public static void main(String[] args) {
7          // TODO Auto-generated method stub
8          int m, n, p, q, sum = 0, c, d, k;
9
10           Scanner in = new Scanner(System.in);
11           System.out.println("Enter the number of rows and columns of first m
12           m = in.nextInt();
13           n = in.nextInt();
14
15           int first[][] = new int[m][n];
16
17           System.out.println("Enter the elements of first matrix");
18
19           for ( c = 0 ; c < m ; c++ )
20              for ( d = 0 ; d < n ; d++ )
21                 first[c][d] = in.nextInt();
22
23           System.out.println("Enter the number of rows and columns of second
24           p = in.nextInt();
25           q = in.nextInt();
26
```

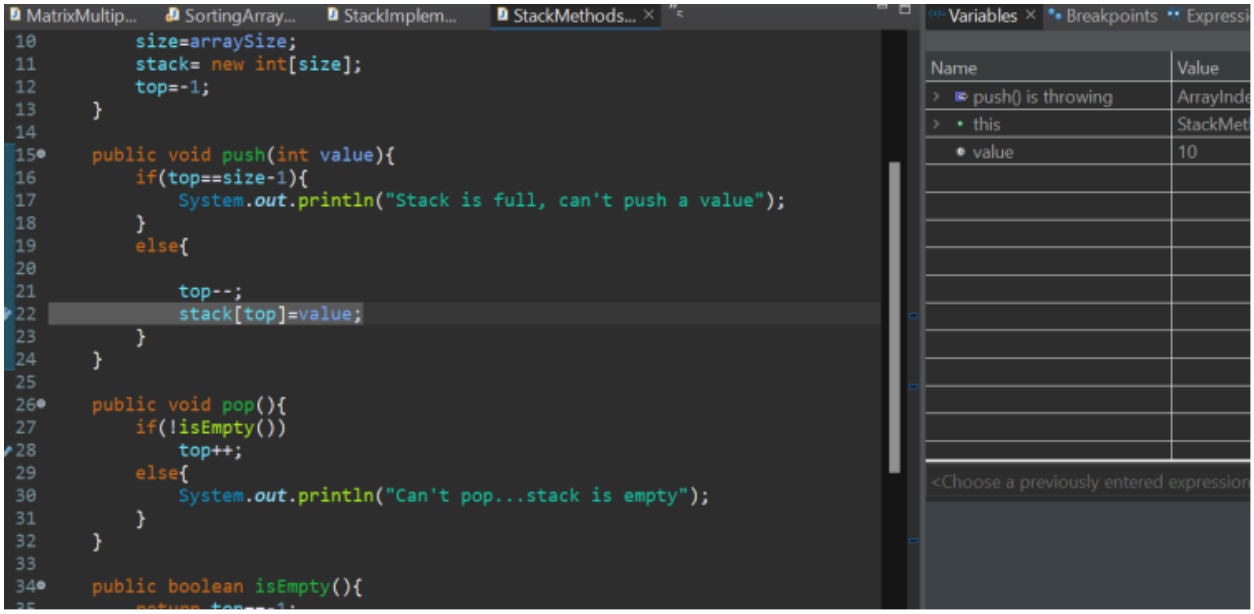
```java
            q = in.nextInt();

            if ( n != p )
               System.out.println("Matrices with entered orders can't be multip
            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 < p ; k++ )
                     {
                        sum = sum + first[c][k]*second[k][d];
                     }

                     multiply[c][d] = sum;
```

```java
                     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");
               }
            }

      }

}
```

Console × Problems Debug Shell

```
<terminated> MatrixMultiplication [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 6:40:12 pm – 6:40:26 pm) [p
1 0
1 0
Product of entered matrices:-
3       0
7       0
```

## 8. Stack Implementation




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

There were 2 errors in this program. The for loop condition at the end for printing, as well as the changes in the top variable when implementing stack pop or push.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With the line breaks at the end of the push, pop and in the loop of the display functions, we can iteratively observe the values of top, and how the referenced value changes. Once we figure out the calculation error, we can simply update the top variable correctly.

**Question 3. Submit your complete executable code.**

```java
package DebugStackImplementation;

public class StackImplementation {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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();
    }

}
```

Console × Problems Debug Shell

```
<terminated> StackImplementation [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 7:14:01 pm – 7:14:04 pm) [pid: 290
10 1 50 20 90
10
```

MatrixMultip... SortingArray... StackImplem... StackMethods... × Variables ×

```java
package DebugStackImplementation;

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;
        }
    }
```

```java

    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();
    }
}

```

## 9. Tower of Hanoi

```java
package DebugTowerOfHanoi;

public class TowerOfHanoi {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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++, inter, from, to);
        }
    }

}
```

| Name | Value |
|---|---|
| no method return value | |
| topN | 2 |
| from | C |
| inter | A |
| to | B |

<Choose a previously entered expression>

Console × Problems Debug Shell

```
owerOfHanoi [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 11:13:13 pm) [pid: 3972]
Disk 1 from A to C
Disk 2 from A to B
Disk 1 from C to A
Disk 2 from C to B
```

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

After removing the syntax errors, there was only one error in the code. The value for topN in the second recursion call was increasing, whereas it should decrease and the recursion would take care of the code. Hence, with that one change, the code works perfectly.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

With two break point on the 19th and 20th line, we were able to figure out that the output of the code was not coming to be correct after the 2nd iteration. That's when it was possible to realize that the error is in the second recursion call, mainly due to the topN value being incremented.

**Question 3. Submit your complete executable code.**

```java
1 package DebugTowerOfHanoi;
2
3 public class TowerOfHanoi {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int nDisks = 3;
8         doTowers(nDisks, 'A', 'B', 'C');
9     }
10
11    public static void doTowers(int topN, char from,
12                char inter, char to) {
13
14        if (topN == 1){
15            System.out.println("Disk 1 from " + from + " to " + to);
16        }
17        else{
18            doTowers(topN - 1, from, to, inter);
19            System.out.println("Disk " + topN + " from " + from + " to " + to);
20            doTowers(topN - 1, inter, from, to);
21        }
22    }
23
24 }
```

Console × Problems Debug Shell

```
<terminated> TowerOfHanoi [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 11:17:33 pm – 11:17:33 pm) [pid: 12124]
Disk 1 from A to C
Disk 2 from A to B
Disk 1 from C to B
Disk 3 from A to C
Disk 1 from B to A
Disk 2 from B to C
Disk 1 from A to C
```

## 10. Quadratic Probing

```
            }

            //i += (i + h / h--) % maxSize;
            i += (i + h * h) % maxSize;

        } while (i != tmp);

    }

    /** Function to get value for a given key **/
    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;
            System.out.println("i "+ i);
        }
        return null;
    }
```

```
Console × Problems Debug Shell
QuadraticProbing [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 11:51:19 pm) [pid: 28852]
3. get
4. clear
5. size
2
Enter key
d
```

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

There were 3 errors in the code, which were mostly based on the logical implementations of the inset, get and remove statements. This can be resolved using counter counters in the h variable.

**Question 2. How many breakpoints you need to fix those errors? What are the steps you have taken to fix the error you identified in the code fragment?**

The errors can be fixed by adding the debugging points on each and every function call in the hash table class.

**Question 3. Submit your complete executable code.**

```java
package DebugQuadraticProbing;
import java.util.Scanner;

public class QuadraticProbing {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        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() ); /** maxSizeake object of Quadrat



        char ch;

        /**  Perform QuadraticProbingHashTable operations  **/
        do{
            System.out.println("\nHash Table Operations\n");
            System.out.println("1. insert ");
            System.out.println("2. remove");
            System.out.println("3. get");
```

```java
            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;
```

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

Console × Problems Debug Shell

```
<terminated> QuadraticProbing [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (20 Oct 2024, 11:44:43 pm – 11:45:13 pm) [pid: 32196]
5. size
1
Enter key and value
d desktop

Hash Table:
d desktop
c computer
```

MatrixMultip... SortingArray... × StackImplem... StackMethods... TowerOfHano... QuadraticPro... QuadraticPro... ×

```java
package DebugQuadraticProbing;

public class QuadraticProbingHashTable {

    private int currentSize, maxSize;
    private String[] keys;
    private String[] vals;

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

    /** Function to clear hash table **/
    public void makeEmpty()
    {
        currentSize = 0;
        keys = new String[maxSize];
        vals = new String[maxSize];
```

```java
    }

    /** Function to get size of hash table **/
    public int getSize()
    {
        return currentSize;
    }

    /** Function to check if hash table is full **/
    public boolean isFull()
    {
        return currentSize == maxSize;
    }

    /** Function to check if hash table is empty **/
    public boolean isEmpty()
    {
        return getSize() == 0;
    }

    /** Function to check if hash table contains a key **/
    public boolean contains(String key)
    {
        return get(key) !=  null;
```


```java
    /** Function to check if hash table contains a key **/
    public boolean contains(String key)
    {
        return get(key) !=  null;
    }
```


```java

    /** Function to get hash code of a given key **/
    private int hash(String key)
    {
        return key.hashCode() % maxSize;
    }

    /** Function to insert key-value pair **/
    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;
```