package Day12;

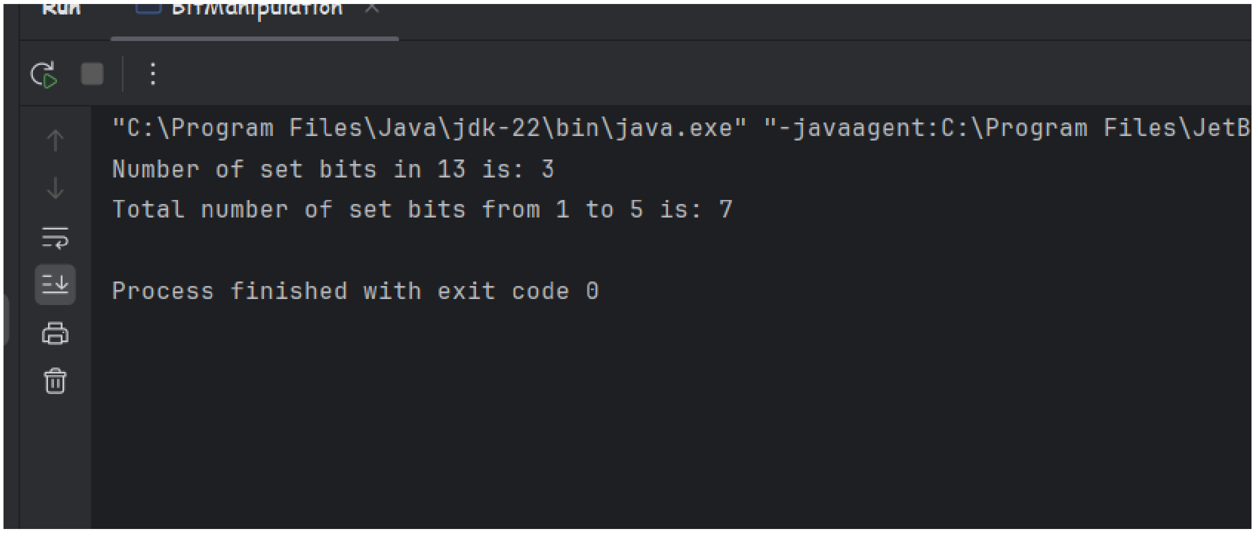
Task 1: Bit Manipulation Basics

Create a function that counts the number of set bits (1s) in the binary representation of an integer. Extend this to count the total number of set bits in all integers from 1 to n.

Ans:

```
public class BitManipulation {

    public static int countSetBits(int num) {
        int count = 0;
        while (num > 0) {
            count += (num & 1);
            num >>= 1;
        }
        return count;
    }

    public static int countTotalSetBits(int n) {
        int totalSetBits = 0;
        for (int i = 1; i <= n; i++) {
            totalSetBits += countSetBits(i);
        }
        return totalSetBits;
    }

    public static void main(String[] args) {
        int num = 13;
        System.out.println("Number of set bits in " + num + " is: " + countSetBits(num));

        int n = 5;
        System.out.println("Total number of set bits from 1 to " + n + " is: " + countTotalSetBits(n));
    }
}
```

Output:



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetB
Number of set bits in 13 is: 3
Total number of set bits from 1 to 5 is: 7

Process finished with exit code 0
```

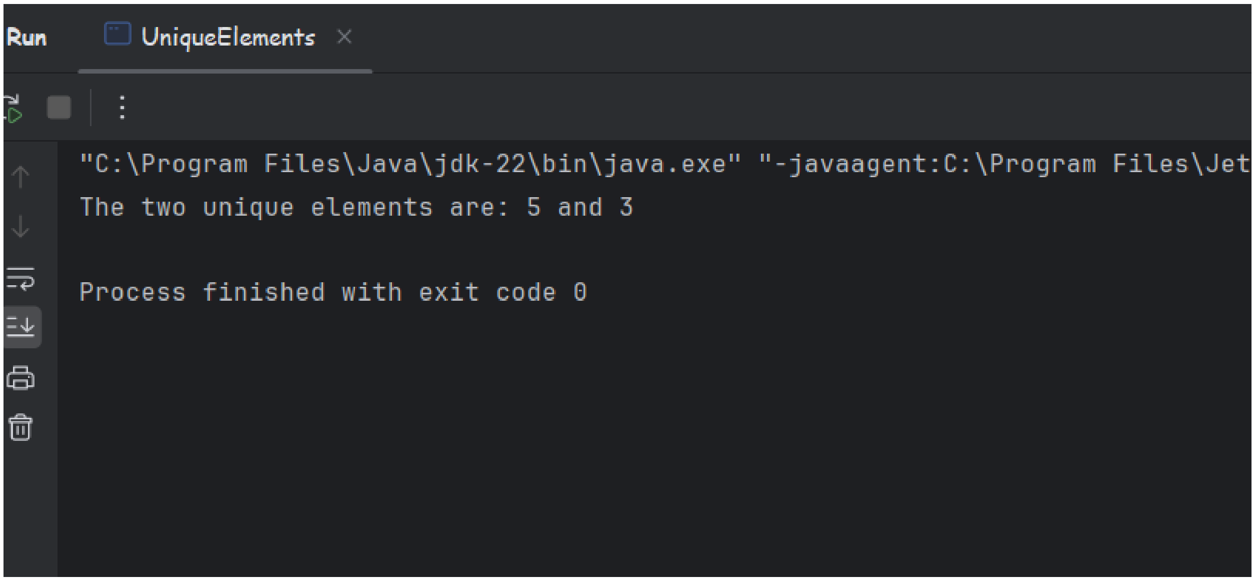
Task 2: Unique Elements Identification

Given an array of integers where every element appears twice except for two, write a function that efficiently finds these two non-repeating elements using bitwise XOR operations.

Ans:

```
package Day12;

public class UniqueElements {

    public static int[] findUniqueElements(int[] nums) {

        int xor = 0;
        for (int num : nums) {
            xor ^= num;
        }

        int setBit = xor & -xor;

        int[] result = new int[2];
        for (int num : nums) {
            if ((num & setBit) == 0) {
                result[0] ^= num;
            } else {
                result[1] ^= num;
            }
        }

        return result;
    }

    // Main method to test the function
    public static void main(String[] args) {
        int[] nums = {1, 2, 1, 3, 2, 5};
        int[] uniqueElements = findUniqueElements(nums);
        System.out.println("The two unique elements are: " + uniqueElements[0] + " and " +
uniqueElements[1]);
    }
}
```

Output:



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\Jet
The two unique elements are: 5 and 3

Process finished with exit code 0
```