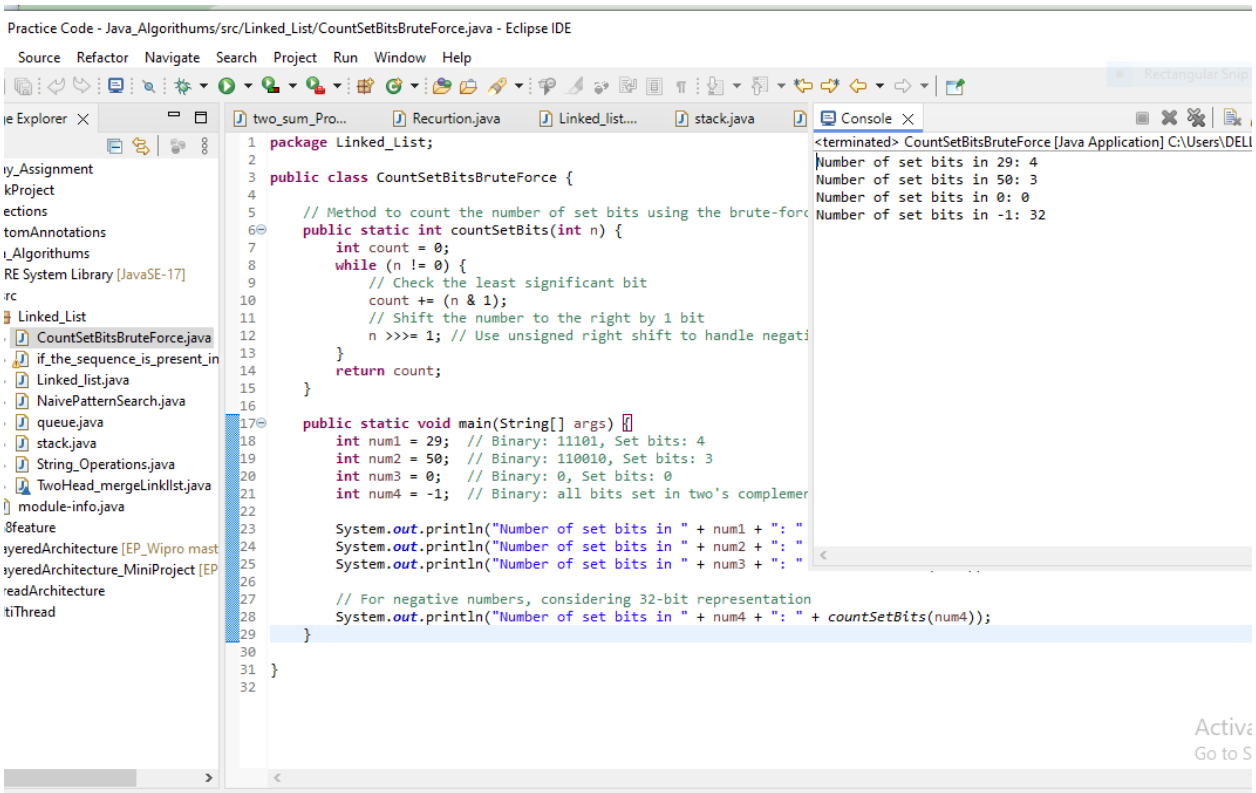
Day 12:

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.

- **Process**:
  - Initialize a counter count to 0.
  - Use a while loop to iterate as long as n is not zero.
  - In each iteration, check the least significant bit (LSB) of n using n & 1.
    - If the LSB is 1, increment the count.
  - Right shift n by one position using n >>>= 1. The >>> operator performs an unsigned right shift, filling the leftmost bit with zero. This is particularly useful for handling negative numbers correctly.
- **Output**: Return the count which represents the number of set bits.



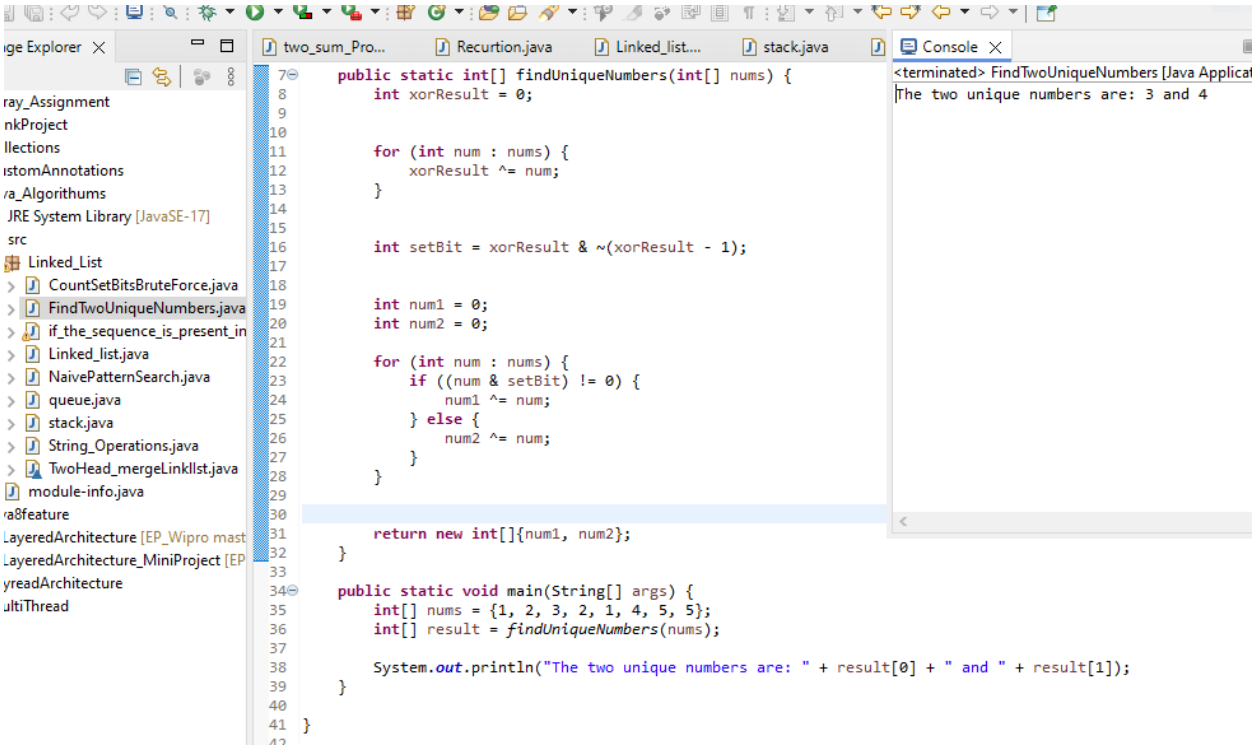
Task 2: Unique Elements Identification.

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.

To find the two non-repeating elements in an array where every other element appears twice, you can use the properties of the XOR operation. This approach takes advantage of the fact that XOR-ing a number with itself results in 0 and XOR-ing a number with 0 results in the number itself.

Here is the efficient algorithm using bitwise XOR operations:

1. **XOR all elements in the array**: This step will give you the XOR of the two unique numbers because the duplicate numbers will cancel each other out.
2. **Find a set bit in the result**: This bit is set in one of the unique numbers and not in the other.
3. **Divide the array into two groups**: One group where the set bit is present and another where it is not.
4. **XOR all elements within each group**: This will isolate the two unique numbers.


ge Explorer
ray_Assignment
nkProject
llections
stomAnnotations
va_Algorithums
JRE System Library [JavaSE-17]
src
Linked_List
CountSetBitsBruteForce.java
FindTwoUniqueNumbers.java
if_the_sequence_is_present_in
Linked_list.java
NaivePatternSearch.java
queue.java
stack.java
String_Operations.java
TwoHead_mergeLinkllst.java
module-info.java
va8feature
LayeredArchitecture [EP_Wipro mast
LayeredArchitecture_MiniProject [EP
yreadArchitecture
ultiThread
two_sum_Pro...
Recurtion.java
Linked_list....
stack.java
Console
<terminated> FindTwoUniqueNumbers [Java Applicat
The two unique numbers are: 3 and 4
public static int[] findUniqueNumbers(int[] nums) {
int xorResult = 0;
for (int num : nums) {
xorResult ^= num;
}
int setBit = xorResult & ~(xorResult - 1);
int num1 = 0;
int num2 = 0;
for (int num : nums) {
if ((num & setBit) != 0) {
num1 ^= num;
} else {
num2 ^= num;
}
}
return new int[]{num1, num2};
}
public static void main(String[] args) {
int[] nums = {1, 2, 3, 2, 1, 4, 5, 5};
int[] result = findUniqueNumbers(nums);
System.out.println("The two unique numbers are: " + result[0] + " and " + result[1]);
}
}