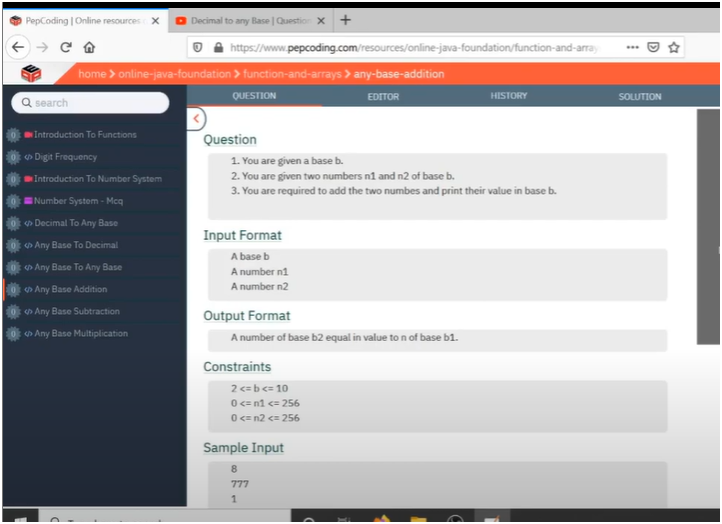
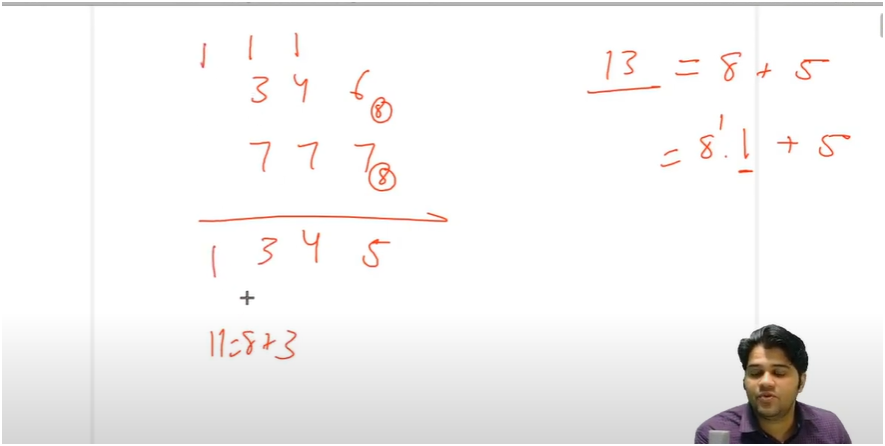
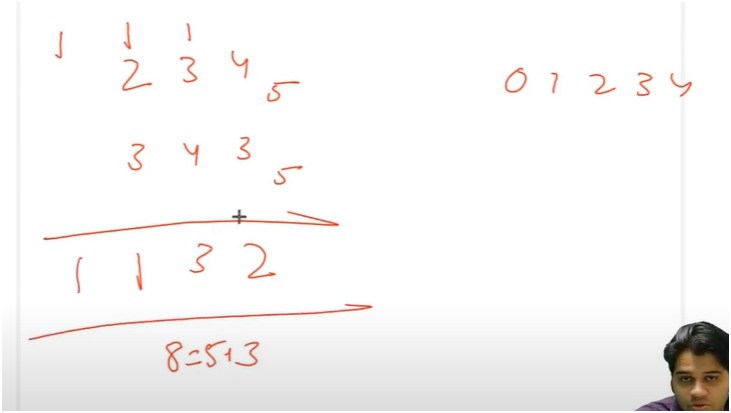
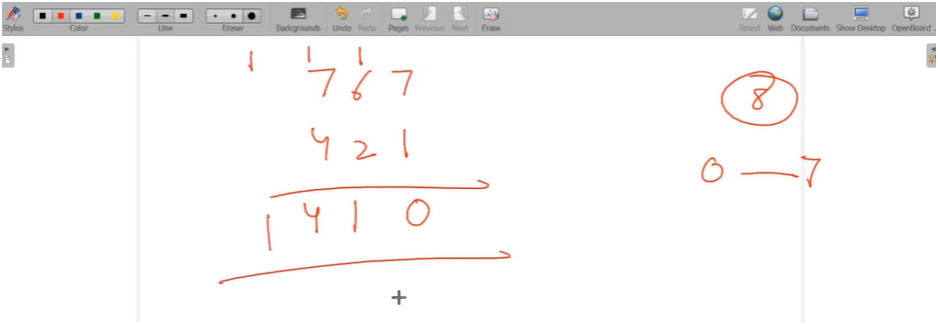
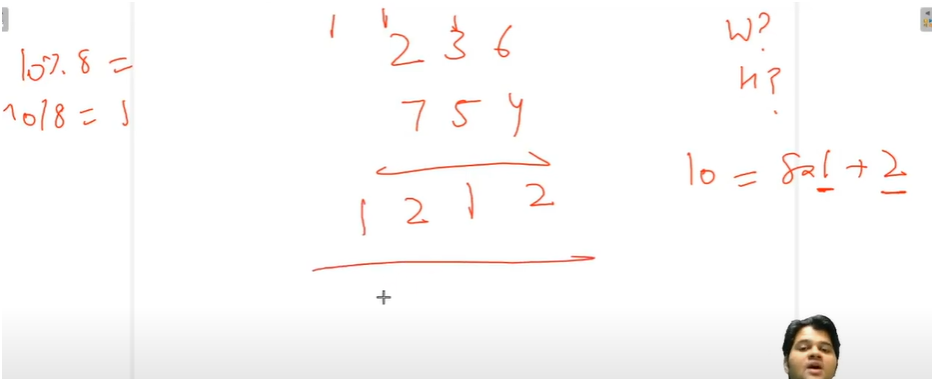
NUMBER BASE ADDITION



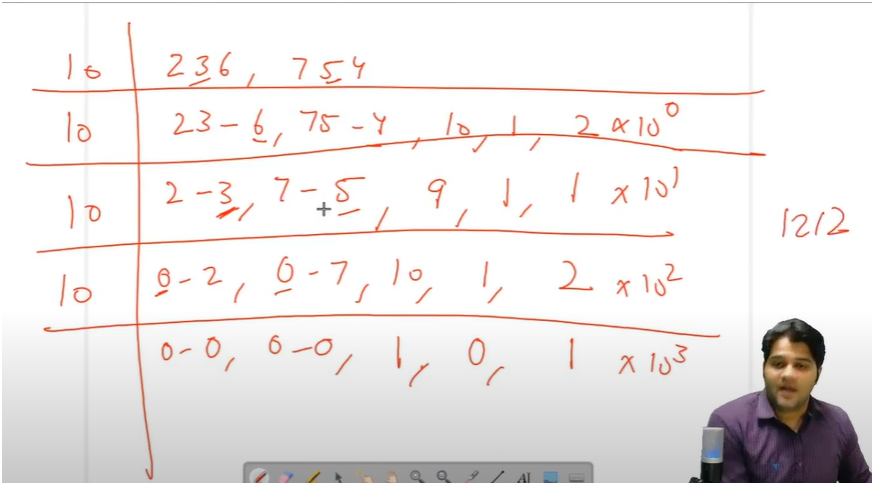


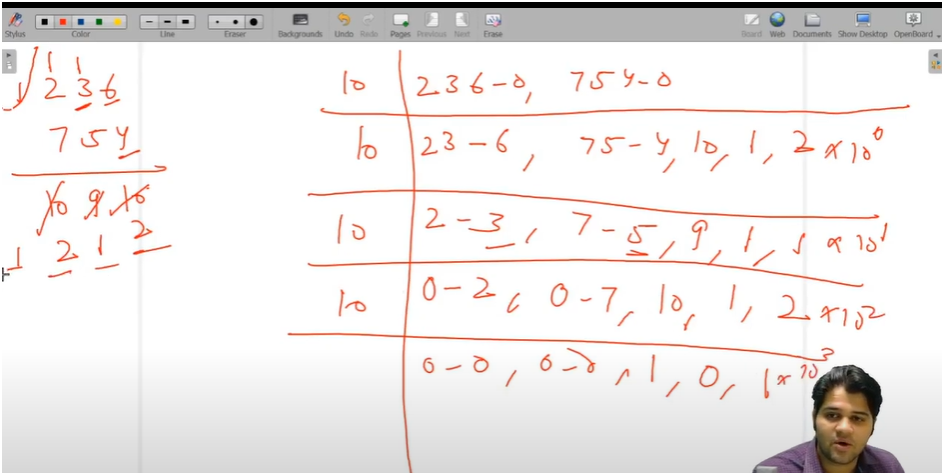






**10|NUMBER1,NUMBER2|ADDITION\_REMAINDERS|DIVDE BY BASE(CARRY)|MODULUS BY BASE(RESULT)**





**TIPS:- Digit + Digit + Carry, Mod by Base, Carry Forward, and Multiply by Powers of 10**

**UNDERSTANDING:-**

**IF U OBSERVED CAREFULLY IT IS ADDITION TWONUMBERS WITH SAME BASE**

* **HERE WE NEED TO RUN A WHILE LOOP UNTIL CARRY REAMINDER AND QUOTIENT DOESNOT BECOMES ZERO**
* **THEN CONTINUOUS MODULUS FIRST AND SECOND NUMBER THEN STORE EACH REMAINDER AND THEN ADD BOTH THE REMAINDERS AND STORE IT IN SUM VARAIBLE** 
  + **NOTE AS WELL AS DON’T FORGET TO ADD CARRY IN IT**
* **NOW CARRY IS CALCULATED BY DIVIDING THE SUM AND GET ITS QUOTIENT AND ALSO MODULUS THE SUM AND GET ITS REMAINDER NOW STORE IT IN VARAIBLE CARRY AND DIGIT**
* **THEN MULTIPLY THE DIGIT WITH INCREASING POWERS OF 10 TO PRINT DIGITS IN REVERSE ORDER**

**ALGORITHM**

**✅ Concept: Addition of Two Numbers in the Same Base**

**✅ Step-by-Step:**

**Initialize:**

**carry = 0**

**power = 1 (to build final number in correct order)**

**sum = 0 (final result)**

**Run a While Loop:**

**Loop continues until all of these are 0:  
num1, num2, and carry**

**while (n1 != 0 || n2 != 0 || carry != 0)**

**Inside Loop:**

**Extract digits (from right to left):  
d1 = n1 % 10  
d2 = n2 % 10**

**Remove used digits:  
n1 = n1 / 10  
n2 = n2 / 10**

**Calculate digitSum:  
digitSum = d1 + d2 + carry**

**Update carry and result digit:**

**digit = digitSum % base**

**carry = digitSum / base**

**Add to final result with power of 10:  
sum += digit \* power**

**Update power: power \*= 10**

**✍️ TIP TO REMEMBER:**

**WHILE (num1 ≠ 0 || num2 ≠ 0 || carry ≠ 0):**

**1. Take remainder of both numbers**

**2. Add them with carry**

**3. Modulo by base → digit**

**4. Divide by base → new carry**

**5. Build result with powers of 10**

**🧠 One-Liner Memory Trick:**

**“Digit + Digit + Carry, Mod by Base, Carry Forward, and Multiply by Powers of 10”**

**Would you like this concept implemented in a Java method too?**