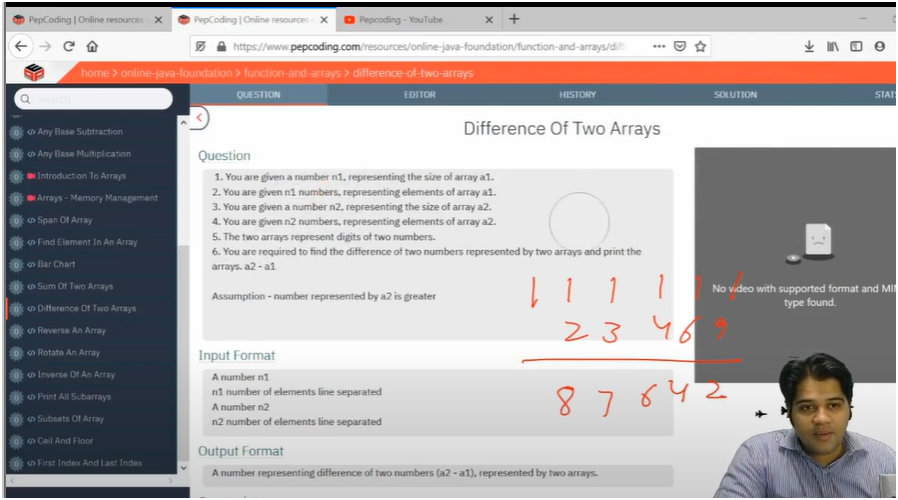
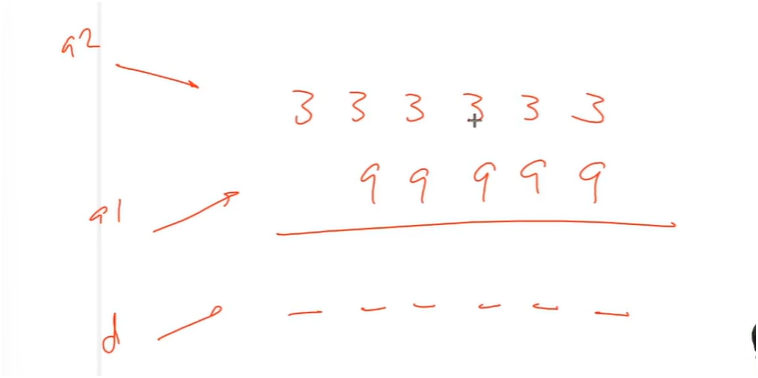
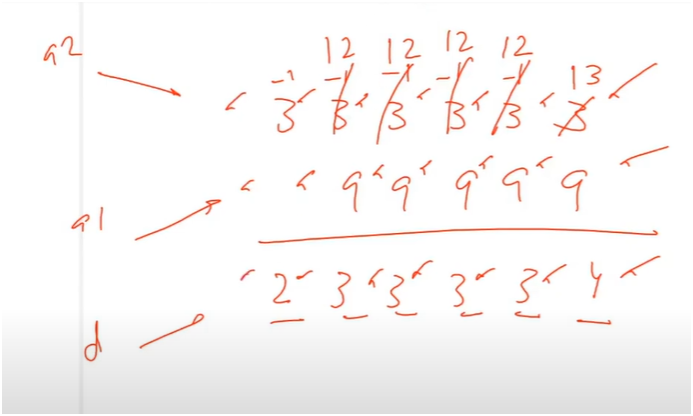
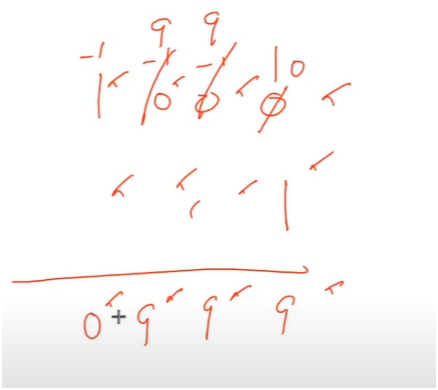
DIFFERENCE OF TWO ARRAYS





SEE 3-9 IS NOT POSSIBLE THERE FORE MAKE 3 AS 13 AND TAKE BORROW -1 ON 3 NOW SEE -1+3-9 IS NOT POSSIBLE THERFORE MAKE BORROW 12 AND THEN 12-9=3 AND SO ON





**UNDERSTANDING(HERE ARRAY2 SIZE MUST BE GREATER THAN ARRAY1 SIZE)**

* **FIRST TAKE TWO ARRAYS AND ASSIGN VALUES TO IT**
* **THEN NOW DEFINE THE SIZE FOR RESULT ARRAY**
  + **SIZE OF THE RESULT ARRAY MAINLY DEPENDS UPON THE LARGEST ARRAY SIZE**
* **THEN INITILAIZE I,j,k TO LAST POSITION OF ARRAY**
* **THEN RUN WHILE LOOP UNTIL K>=0** 
  + **NOW IF IF ELEMENT(1ST ARRAY) + BORROW > ELEMENT(2ND ARRAY)**
    - **THEN ELEMENT(1ST ARRAY) + BORROW - ELEMENT(2ND ARRAY) STORED IT IN VARAIBLE D**
    - **NO BORROW**
  + **IF ELEMENT(1ST ARRAY) + BORROW < ELEMENT(2ND ARRAY)** 
    - **ELEMENT(1ST ARRAY) + BORROW + 10 - ELEMENT(2ND ARRAY) AND STORED IT IN VARAIBLE D**
    - **THEN TAKE A BORROW OF -1**
  + **BUT ALSO WRITE A CHECK WHEN ONE ARRAY IS BIGGER THAN ANOTHER ARRAY**
    - **IF ONE ARRAY IS SMALLER THEN ADD ZEROS FROM LEFT SIDE**
  + **NOW STORE THE VARIBALE D IN RESULT ARRAY**
  + **THEN DECREMENT I ,j ,k**
* **BUT ALSO WE NEED TO ADD ONE CHECK**
  + **SUCH AS 00091 IS COUNT AS 91**
  + **AND 1001 IS COUNT AS 1001**
* **THEN AT THAT TIME INTIALIZE INDEX TO 0**
  + **AND THEN RUN A WHILE LOOP COMPARING INDEX WITH RESULT ARRAY LENGTH**
  + **IF RESULT ARRAY INDEX IS ZERO THEN INCREMENT INDEX**
  + **ELSE BREAK**
* **AND THEN PRINT RESULT ARRAY**

**ALGORITHM:-**

**🧾 Step-by-Step Logic**

**🔹 1. Input Two Arrays**

* **Create two arrays to store digits of both numbers.**
* **Assign values to the arrays from most significant to least significant digit.**

**🔹 2. Prepare the Result Array**

* **Result array size should be equal to the maximum size of the two arrays (since subtraction never increases the number of digits).**
* **Initialize a result array of that size.**

**🔹 3. Initialize Pointers**

* **Use three variables:**
  + **i → points to last index of the first array**
  + **j → points to last index of the second array**
  + **k → points to last index of the result array**

**🔁 4. Loop: While k >= 0**

**At each step:**

**🔸 A. Handle shorter array**

* **If one array is shorter than the other, assume leading zeros.**
  + **Example: 0056 – 123 → pad 56 as 0056**

**🔸 B. Perform Subtraction with Borrow:**

* **If: second[j] + borrow >= first[i]**
  + **Then: d = second[j] + borrow - first[i]**
  + **Set borrow = 0**
* **Else:**
  + **Add 10: d = second[j] + borrow + 10 - first[i]**
  + **Set borrow = -1**

**🔸 C. Store Result Digit:**

* **Place the value d in result[k]**

**🔸 D. Decrement Pointers:**

* **i--, j--, k--**

**🔍 5. Remove Leading Zeros from Result**

* **Initialize an index = 0**
* **While result[index] is 0, increment index**
* **Break when a non-zero digit is found**

**Example:  
Result array = [0, 0, 0, 9, 1] → should be printed as 91**

**📤 6. Print the Final Answer**

* **Print from result[index] till the end of the array**