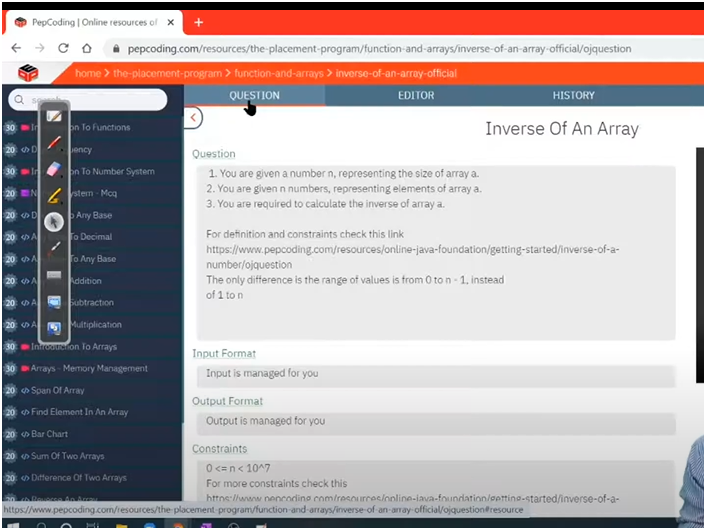
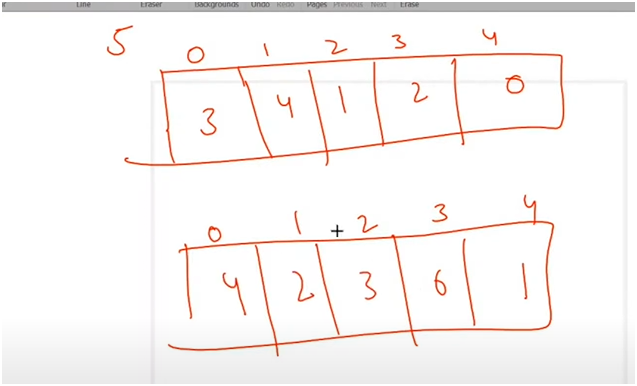
INVERSE OF AN ARRAY

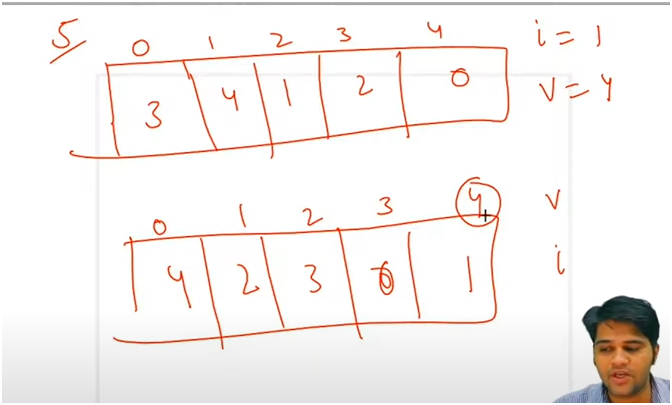
**🎯 Goal:**

To create a new array (called the **inverse array**) such that:

If the original array has a value val at index i,  
Then the inverse array will have a value i at index val.







**UNDERSTANING:-**

* **SIMPLE TRICK** 
  + **ORIGINAL ARRAY MAE JO VALUE AHE USKO INVERSE ARRAY MAE INDEX BANAO**
  + **AND ORIGINAL ARRAY MAE JO INDEX AHE USKO INVERSE ARRAY MAE VALUE BANAO**
* **IF WE OBSERVED CAREFULLY WHATEVER VALUES PRESENT IN THE ARRAY IS BETWEEN THE INDEX 0 TO N** 
  + **MEANS IF N=4 THEN IN ARRAY VALUES MUST BE BETWEEN 0 TO 3**
* **FIRST GET ARRAY AND INSERT VALUES IN IT** 
  + **THEN U OBSERVE CAREFULLY FIRST ARRAY AND INVERSE ARRAY**
  + **WHATEEVER VALUES PRESENT AT IN INDEX IS INVERSE IN INVERSE ARRAY**
  + **FIRST ARRAY**
    - **INDEX=0 VALUE=3**
    - **INDEX=1 VALUE=4**
    - **INDEX=2 VALUE=1**
    - **INDEX=3 VALUE=2**
    - **INDEX=4 VALUE=0**
  + **INVERSE ARRAY**
    - **INDEX=3 VALUE=0**
    - **INDEX=4 VALUE=1**
    - **INDEX=1 VALUE=2**
    - **INDEX=2 VALUE=3**
    - **INDEX=0 VALUE=4**
* **THUS RUN A FOR LOOP TO TRAVERSE EACH ELEMENT IN AN ARRAY**
  + **AND THEN AT VALUE ASSIGN ARRAYS INDEX VALUE**
  + **AND AT INDEX ASSIGN ARRAYS INDEX**
* **AND THEN RETURN ARRAY**

**ALGORITHM:-**

**🧾 Step-by-Step Logic:**

1. **Input the Array**
   * **Read the size n.**
   * **Fill the array of size n with values ranging from 0 to n-1.**
2. **Initialize an Inverse Array**
   * **Create a new array of the same size as the original.**
3. **Loop Through the Original Array**
   * **For each index i, get the value arr[i].**
   * **In the inverse array, set:**

**At index = arr[i], store value = i**

1. **Return or Print the Inverse Array**