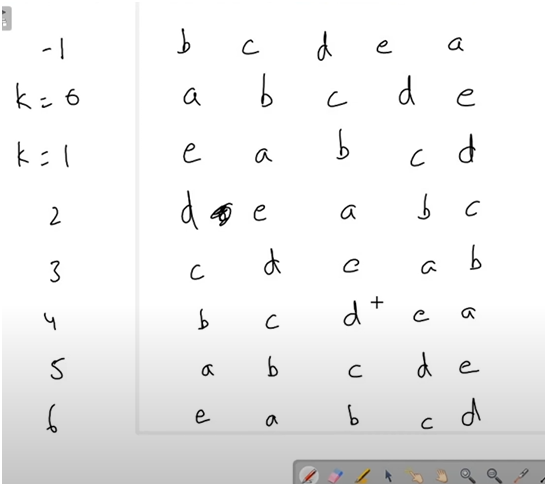
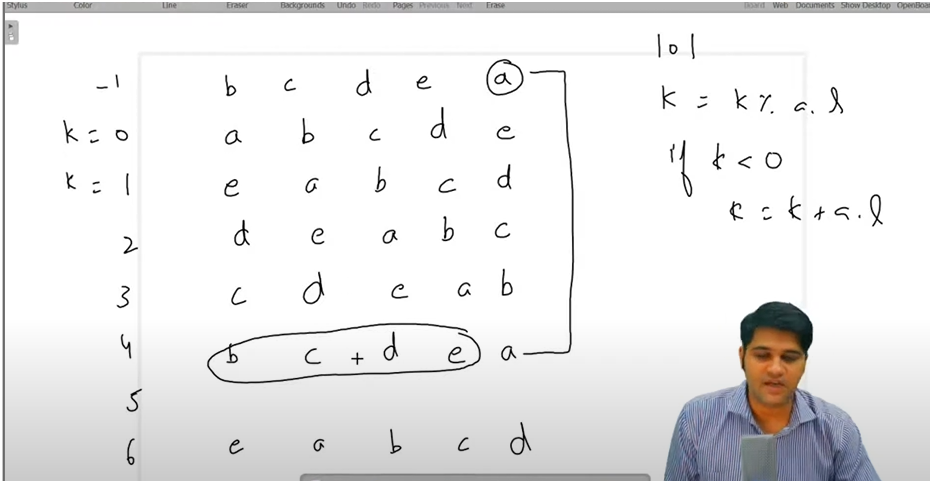


NEGATIVE AND POSITIVE ROTATING





AND IF WE OBSERVE AFTER EVERY 5 ROTATIONS WE GET ORIGINAL STRING AGAIN

FOR 11 ROTATION means 11%10=1 DOES WEE NEED TO DO JUST 1 ROTATION MEANS 11 ROTATION IS EQL TO 1 ROTATION

**I=0 a b c d e**

**I=1 e a b c d**

**I=2 d e a b c**

**I=3 c d e a b**

**I=4 b c d e a**

**I=5 a b c d e**

**I=6 e a b c d**

**I=7 d e a b c**

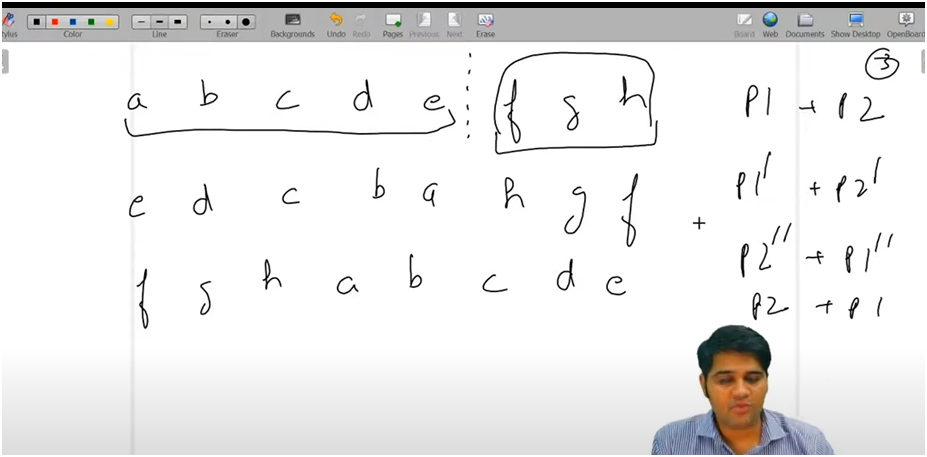
**I=8 c d e a b**

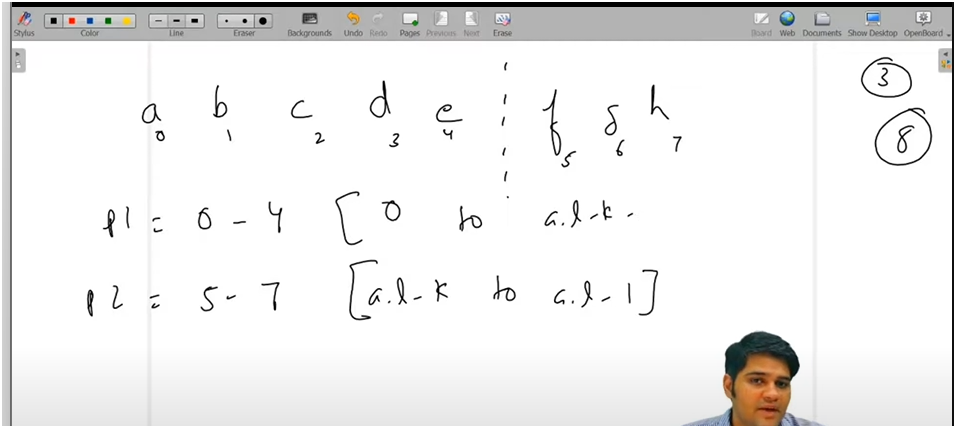
**I=9 b c d e a**

**I=10 a b c d e**

**I=11 e a b c d**

AS K=-1 IS EQL TO K=4 ROTATIONS





**UNDERSTANDING:-**

**IN SIMPLE WORDS DIVIDE THE STRING INTO TWO PARTS THEN APPLY REVERSE TO EACH PART**

* **ROTATE FUNCTION**
  + **FIRST HANLE NEGATIVE ROTATION**
  + **NEGTIVE ROTATION** 
    - **FOR NEGATIVE ROTATION WE JUST ADD NEGTIVE ROTATION TO ARRAYS LENGTH**
    - **MEANS**
      * **AS K=-1 IS EQL TO K=4 ROTATIONS**
  + **AND FOR POSTIVE ROTATION**
    - **TOTAL ROATION MODULUS 10**
    - **MEANS**
      * **AND IF WE OBSERVE AFTER EVERY 5 ROTATIONS WE GET ORIGINAL STRING AGAIN**
      * **FOR 11 ROTATION means 11%10=1 DOES WEE NEED TO DO JUST 1 ROTATION MEANS 11 ROTATION IS EQL TO 1 ROTATION**
  + **FIRST PART WILL START FROM 0 TO ARRAYS LENGTH-ROTATION NUMBER-1**
    - **THEN PASS ARRAY AND START AND LAST POSTION TO REVERSE FUNCTION**
  + **SECOND PART WILL START FROM ARRAYS LENGTH -ROTATION TO ARRAYS LENGTH-1**
    - **SIMILARLY PASS ARRAY AND START AND LAST POSTION TO REVERSE FUNCTION**
* **REVERSE FUNCTION**
  + **FOLLOWS TWO POINTER APPROACH**
    - **FIRST POINTER POINTS TO STARTING POS AND SECOND POINTER TO END POS**
    - **PERFORM SWAP LOGIC**
    - **THEN INCREEMENT FIRST POINTER AND DECREMENT SECOND POINTER**

**NOTE:-BEFORE PASSING TO THE POS TO REVERSE FUNCTION ATTACH A CHECK FOR NEGATIVE ROTATION AND**

**POSITVE ROTATION**