Explaination

**Question-1** :

Step 1 : Convert inputString to charArray

Step 2: Initialization

Step 3: Creating LinkedHashMap with characters as keys and their position as values

Step 4: Iterating through charArray.

Step 5: If ch is not present in charPosMap, adding ch into charPosMap along with its position.

Step 6: If ch is already present in charPosMap, reposioning the cursor i to the position of ch and clearing the charPosMap.

Step 7: Updating longestSubstring.

Step 8: Printing the longest Substring.

**Question-2 :**

Step 1: Declare the first number as 0 and second number as 1

Step 2: Declare the third variable which Holds the addition value of first and secong variable

Step 3: Print the first and second variable

Step 4 : Print the third variable using the for loop until condtion is true

**Question-3 :**

Step 1:Function for swapping the characters at position I with character at position j

Step 2: Declaring the variable and giving the value CAT

Step 3: Function for generating different permutations of the string

Step 4: Prints the permutations

Step 5: Swapping the string by fixing a character

Step 6: Recursively calling function generatePermutation() for rest of the characters

Step 7: Backtracking and swapping the characters again

**Question-4 :**

Step 1: Declaring the two string variables with “Welcome “ and “ceelmow”.

Step 2: check if length is same

Step 3: convert strings to char array

Step 4: sort the char array

Step 5: if sorted char arrays are same then the string is anagram

**Question-5 :**

Yes, you can implement a stack using only queues in Java without using arrays or linked lists.

The key idea is to use two queues to simulate the stack operations - push, pop, top.

In this approach, the push operation is implemented by enqueuing elements into one of the queues.

When a pop or top operation is requested, the elements are dequeued from the first queue and enqueued into the second queue until the last element is reached.

Then the last element is either popped or returned as the top element, simulating the stack behaviour. Then to maintain the order of elements, the two queues are swapped after each pop or top operation.

The implementation ensures that the most recently added element is always at the front of the queue, mimicking the behaviour of a stack.

This way, a stack can be effectively realized using only queues in Java, without relying on arrays or linked lists.

**Question-6 :**

Step 1: Declare the String variable with String “Welcome”

Step 2: for Loop- extracts each character, adds each character in front of the existing string

Step 3: Print the reversed String