08/07/2025:

Steps:

Input: abac;

Count[] 🡪 keeps how many time each character has appeared

Queue 🡪 keeps track of characters int order they came.

Result 🡪 The final output stream.

Let the first character be ‘A’. Add ‘A’ to the queue, count of ‘A’ =1🡪 count[A] = 1

Now, front of queue is ‘A’ and it is non-repeating, hence add ‘A’ to result.

STEP2:

Now the second character is ‘B’, add ‘B’ to queue[‘A’,’B’] so now count[‘B‘] = 1

Now front of queue is ‘A’ and count of ‘A’ = 1; ‘A’ is still non-repeating, so we will add it to a result and result = ‘AA’.

STEP3:

Now, the third character is ‘A’ add ‘A’ again to the queue[‘A’,’B’,’A’] so now count[‘A’] =2 and count[‘B’] = 1

Now we will remove front ‘A’ from Queue and queue will be [‘B’,’A’].

Now, front is ‘B’, so count of ‘B’=1 and finally we will add ‘B’ to the result.

Now, the next character is C, so the queue is ‘B’,’A’,’C’, now count of C =1, since front is B, so count of B = 1,now we will add ‘B’ to the result again. The result will be AABB and final answer = AABB

**Summary of what is happening:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Character** | **Queue** | **Count** | **1st non-repeating** | **o/p** |
| A | a | 1 | a | a |
| B | A,b | A:1,b:1 | a | a |
| A | A,b,a 🡪 b,a | A:2,b:1 | b | b |
| C | B,a,c | A:2,b:1,c:1 | b | b |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Character** | **Queue** | **Count** | **1st non-repeating** | **o/p** |
| A | a | 1 | a | a |
| A | A,a🡪a | A:2 | \_\_\_ | # |
| B | A,a,b 🡪 a,b | A:2,b:1 | b | b |
| C | A,a,b,c🡪a,b,c | A:2,b:1,c:1 | b | b |
| D | A,a,b,c,d🡪a,b,c,d | A:2,b:1,c:1,d:1 | b | b |
| D | A,a,b,c,d,d 🡪 a,b,c,d | A:2,b:1,c:1,d:2 | b | b |

Input: aabcdd

**File handing:**

File handling means reading from and writing to files, using a programming language. In Java, file handling help us to store data permanently.

**Why do we need file handling?**

1. **To save user input like notes, logs, reports,**
2. **To read existing data from file.**
3. **To process documents and binary files.**

**Java tools for file handling:**

**Java provides from the java.io package to handle files.**

|  |  |
| --- | --- |
| **Class** | **Purpose** |
| **file** | **To create and manage file/directories** |
| **FileWriter** | **To write character in file** |
| **BufferedReader** | **To read large text files** |
| **FileInputStream** | **To read binary files(image, videos)** |
| **FileOutputStream** | **To write binary files** |

**Two main types of streams:**

**Character streams 🡪 For text files(.txt,.java)**

**Byte stream 🡪 For binary files(.jpg,.pdf)**

**q. Anaya, a curious student who loves coding**

**The File, a magical diary that remembers everything**

**FileWriter, the magic pen**

**FileReader, the magic glasses**

**Java, her trusted guide**

**The Problem**

**One day, Anaya was tired of typing the same notes again and again every time she ran her program. She said:**

**Java, ner ousted guide**

**1 The Problem**

**One day, Anaya was tired of typing the same notes again and again every time she ran her program. She said:**

**Java replied wisely:**

**2. The Magical Dicy (Creating a Filo)**

**Java gave her a magical diary:**

**Problem Filles Log Cleaner-Remove Dupilonte Line**

**Difficulty: Easy to Intermediate**

**Tags: File Handling, HashSet, BufferedReader, BufferedWrite**

**Problem Sisten**

**You are given a text file named "log.txt" that contains multiple lines. Some lines may be**

**duplicates.**

**Write a program to:** **Q 1. Problem Filles Log Cleaner-Remove Duplicate Line.**

**Difficulty: Easy to Intermediate**

**Tags: File Handling, HashSet, BufferedReader, BufferedWrite**

**Problem Statement:**

**• You are given a text file named "log.txt" that contains multiple lines. Some lines may be duplicates.**

**• Write a program to:**

** Read all lines from the log.txt**

** Remove duplicate line**

** Write the cleaned version into a new file called “clean\_log.txt”**

** Maintain the original order of the appearance.**

**Constains:**

**File log.txt exists and is readable**

**Lines can contain alphanumeric.**

**Ask a user for input an output, give it a menu driven structure(Allowing a user to choose what to clean and remove)**

**Option 1 Duplicate line**

**Option 2 Duplicate word**

**Option 3 A particular line**

**Option 4 Last two lines**