EXPERIMENT -3.1(8)

| | |
|---|---|
| NAME-  Jatin | UID- 20BCS5951 |
| BRANCH- BE- CSE | SECTION- DM 605 B |
| SEMESTER- 6$^{th}$ | DATE- 01/05/2023 |
| SUBJECT- CC Lab- 2 | SUBJECT CODE- 20CSP-351 |

Q1. Remove Duplicate Letters

CODE:

```
class Solution {
  public String removeDuplicateLetters(String s) {
    int[] lastIndex = new int[26];
    for (int i = 0; i < s.length(); i++){
      lastIndex[s.charAt(i) - 'a'] = i; // track the lastIndex of character presence
    }

    boolean[] seen = new boolean[26]; // keep track seen
    Stack<Integer> st = new Stack();

    for (int i = 0; i < s.length(); i++) {
      int curr = s.charAt(i) - 'a';
      if (seen[curr]) continue; // if seen continue as we need to pick one char only
      while (!st.isEmpty() && st.peek() > curr && i < lastIndex[st.peek()]){
        seen[st.pop()] = false; // pop out and mark unseen
      }
      st.push(curr); // add into stack
      seen[curr] = true; // mark seen
    }

    StringBuilder sb = new StringBuilder();
    while (!st.isEmpty())
```

```
            sb.append((char) (st.pop() + 'a'));
        return sb.reverse().toString();
    }
}
```

Output :



Q2  Assign Cookies

Code :

```
class Solution {
public int findContentChildren(int[] g, int[] s) {
    Arrays.sort(g);
    Arrays.sort(s);
    int idx = 0, count = 0, len = Math.min(g.length, s.length);
    for (int i = 0; i < len && idx < s.length; i++) count += s[idx++] >= g[i] ? 1 : (--i - i);
    return count;
```

```
    }
}
```

Output :