Kick Start 2022 - Round C

Analysis: New Password

Test Set 1

For this test set, the old password only satisfies requirements 1 and 4. To satisfy other requirements, append a lowercase and an uppercase English alphabet letter and any special character at the end of the old password. These operations can be performed in any order. We can prove that this is the new password with minimum length satisfying all the requirements. The overall complexity of this solution is $O(N)$.

Sample Code (C++)

```
string createNewPassword(string oldPassword) {
    string newPassword = oldPassword;
    newPassword.append("a"); // Append any lowercase English alphabet letter.
    newPassword.append("B"); // Append any uppercase English alphabet letter.
    newPassword.append("&"); // Append any special character.

    return newPassword;
}
```

Test Set 2

To solve this test set, we can loop through the old password and check which of the given requirements are unsatisfied. For each unsatisfied requirements between 2 and 5 (both inclusive), we can just insert respective character and make it satisfied. After performing these operations, if the length of the new password is less than 7, then we can append digits, letters, or special characters until the new password's length becomes 7. We can prove that this is the new password with minimum length satisfing all the requirements. The time complexity of this solution is $O(N)$.

Sample Code (C++)

```
string createNewPassword(string oldPassword) {
    bool condition2 = false;
    bool condition3 = false;
    bool condition4 = false;
    bool condition5 = false;
    string newPassword = oldPassword;
    for (int i = 0; i < oldPassword.size(); i++) {
        if (oldPassword[i] >= 'A' && oldPassword[i] <= 'Z')
          condition2 = true;
        else if (oldPassword[i] >= 'a' && oldPassword[i] <= 'z')
          condition3 = true;
        else if (oldPassword[i] >= '0' && oldPassword[i] <= '9')
          condition4 = true;
        else if (oldPassword[i] == '@' || oldPassword[i] == '#' || oldPassword[i] == '&' || oldPassword[i] =
          condition5 = true;
    }

    if (!condition2) newPassword.append("A"); // Append any uppercase English alphabet letter.
    if (!condition3) newPassword.append("a"); // Append any lowercase English alphabet letter.
    if (!condition4) newPassword.append("1"); // Append any digit.
    if (!condition5) newPassword.append("#"); // Append any special character.

    // Append any digit, letter, or a special character.
    while (newPassword.size() < 7) newPassword.append("1");

    return newPassword;
}
```