Stack-Queue Combined Scenario-Based Problems Worksheet

Problem 1: Airport Passenger Matching

In an airport, domestic (D_id) and international (I_id) passengers must be matched in pairs. The matching rules are as follows:

- A domestic (D) and international (I) traveler appearing consecutively in the stack are immediately paired.
- Consecutive passengers of the same type are queued for future pairing.
- When encountering a passenger with no immediate pair, check the queue for a spare of the opposite type.
- If none found, discard the passenger.

Only use public methods:

- Stack: push(), pop(), peek(), isEmpty()
- Queue: enq(), deq(), peek(), isEmpty()

Given:

```
Stack Top -> D_21, I_33, I_12, I_9, D_5, D_8, D_41, I_15, D_6
```

Output:

```
21 is paired with 33
5 is paired with 12
8 is paired with 9
41 is paired with 15
6 is discarded
```

Problem 2: Vaccine Distribution Center

At a distribution center, vaccine recipients are either elderly (E_id) or frontliners (F_id).

- An elderly and a frontliner appearing consecutively are immediately paired.
- Consecutive elderly or frontliners are queued.
- When encountering a recipient with no match, pair from the queue if possible.
- Otherwise, discard them.

Stack Top -> F_11, E_21, E_33, F_45, F_49, E_88, F_91, F_92

Problem 3: Job Fair Interview Scheduling

Job applicants are of two types: Tech (T_id) and Non-Tech (N_id).

- Pair up T and N if they appear consecutively.
- Queue same-type applicants for future pairing.
- Attempt to pair unmatchable applicants with queued opposite type.
- Discard if no match.

Stack Top -> T_4, N_8, T_6, T_9, N_2, N_7, N_5, T_1, T_3

Problem 4: Animal Shelter Adoption

Animals are either dogs (D_id) or cats (C_id).

- If a dog and cat are next to each other in the stack, adopt them together.
- Otherwise, enqueue spare animals for future pairing.
- Match spare dog with arriving cat, and vice versa.
- Discard if no pair is found.

Stack Top -> C_1, C_4, D_6, D_7, C_10, D_12, C_14

Problem 5: Debate Team Pairing

Debaters are either Pro (P_id) or Con (C_id).

- Pro and Con pairs debate together.
- Same sides are queued.
- Use queue to pair with opposite if direct pairing fails.
- Discard unmatched debaters.

Stack Top -> P_3, C_6, C_7, P_10, P_12, C_13, C_15, P_17

Instructions:

1. Implement each scenario using only stack and queue operations.
2. Do not use arrays, lists, or recursion.
3. Use a helper function to extract the type and ID from each entry (e.g., "P_10" -> type = 'P', id = '10').
4. Print each valid pair and any discarded entries.

End of Worksheet