Given the hash function $h(k)=k \bmod 11$, we need to see if inserting certain keys into the hash table can result in the given table state, with the probing constant $c=1$.

The given hash table is:

| **Index** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Key | 44 | 13 | \_ | \_ | 16 | 24 | 18 | 6 | \_ | \_ | 21 |

Let's go step by step to insert each key and see if it could produce the given table:

1. Insert 44: $44 \bmod 11=0$, so 44 goes into index 0 .

2. Insert 13: $13 \bmod 11=2$, but since index 2 is taken, we probe to the next available slot which is index 1.

3. Insert 16: $16 \bmod 11=5$, which is empty, so 16 goes into index 5 .

4. Insert 24: $24 \bmod 11=2$, but indexes 2 and 3 are taken, so it goes to index 4 .

5. Insert 18: $18 \bmod 11=7$, which is empty, so 18 goes into index 7 .

6. Insert 6: $6 \bmod 11=6$, which is empty, so 6 goes into index 6 .

7. Insert 21: $21 \bmod 11=10$, which is empty, so 21 goes into index 10 .

Now, we need to check if inserting these keys in any order could result in the given hash table.

Since 44 is at index 0 , which is its hash value, it must be inserted first. Next, 13 is at index 1 , which suggests that it was inserted when index 2 was already filled, so it must have been inserted after 24 (since 24 mod 11 is 2 ). Continuing this reasoning, we can deduce the order of the keys inserted to end up with the given table.

Let's proceed to simulate the process and verify the order of insertion that results in the table as given.

The simulation of all possible orders of inserting the keys $[44,13,16,24,18,6,21]$ into the hash table with the hash function $h(k)=k \bmod 11$ and linear probing with $c=1$ did not result in the given table state. This means that the given hash table cannot be the result of inserting the keys with this method.

Let's analyze why this might be the case:

- Key 44 is placed at index 0 , which is correct since $44 \bmod 11=0$.

- Key 13 should be at index 2 , since $13 \bmod 11=2$, but it's at index 1 . This implies that when 13 was inserted, index 2 was already filled.

- Key 16 should be at index 5 , since $16 \bmod 11=5$, which is correct.

- Key 24 should be at index 2 , since $24 \bmod 11=2$, but it's at index 4 . This suggests that when 24 was inserted, indices 2 and 3 were already filled.

- Key 18 should be at index 7 , since $18 \bmod 11=7$, which is correct.

- Key 6 should be at index 6 , since $6 \bmod 11=6$, which is correct.

- Key 21 should be at index 10 , since $21 \bmod 11=10$, which is correct.

From this analysis, it's clear that the positions of keys 13 and 24 are problematic. The key 24 should have been placed before key 13 , as it would naturally fall into the second slot, and 13 would have been probed into the next available slot, which would be 1. Since the simulation could not find a correct order, this confirms that the given table state cannot be a result of inserting these keys using the described linear probing method.