Let's go step by step and see how the operation checker |= (1 << val) works for the string "helol".

We will track the binary value of checker as we process each character:

Step 1: Initialize

- **checker** starts as 00000000000000000000000000 (in binary), which means all bits are set to 0.
- This checker variable will store information about which characters have been seen.

Process each character:

Character 1: 'h'

- 'h' has an ASCII value of 104.
- The position val is calculated as:
  val = 'h'.charCodeAt(0) - 'a'.charCodeAt(0) = 104 - 97 = 7.
  So, val = 7, meaning 'h' corresponds to the 7th bit.
- **1 << val**:
  1 << 7 shifts 1 to the 7th position, so we get:
  00000000000000000010000000.
- **Update checker**:
  Now, checker |= (1 << val) updates the checker to set the 7th bit:

sql

Copy code

```
checker:    00000000000000000000000000  (initial value)

1 << 7:     00000000000000000010000000

---------------------------------------

result:     00000000000000000010000000  (7th bit is set)
```

After processing 'h', checker = 00000000000000000010000000.

Character 2: 'e'

- 'e' has an ASCII value of 101.

- The position val is:
  val = 'e'.charCodeAt(0) - 'a'.charCodeAt(0) = 101 - 97 = 4.
  So, val = 4, meaning 'e' corresponds to the 4th bit.
- **1 << val**:
  1 << 4 shifts 1 to the 4th position, so we get:
  00000000000000000000010000.
- **Update checker**:
  Now, checker |= (1 << val) updates the checker to set the 4th bit:

sql

Copy code

```
checker:    00000000000000000010000000  (from 'h')

1 << 4:     00000000000000000000010000

--------------------------------------

result:     00000000000000000010010000  (7th and 4th bits are set)
```

After processing 'e', checker = 00000000000000000010010000.

Character 3: 'l'

- 'l' has an ASCII value of 108.
- The position val is:
  val = 'l'.charCodeAt(0) - 'a'.charCodeAt(0) = 108 - 97 = 11.
  So, val = 11, meaning 'l' corresponds to the 11th bit.
- **1 << val**:
  1 << 11 shifts 1 to the 11th position, so we get:
  00000000000000100000000000.
- **Update checker**:
  Now, checker |= (1 << val) updates the checker to set the 11th bit:

sql

Copy code

```
checker:    00000000000000000010010000  (from 'h' and 'e')

1 << 11:    00000000000000100000000000

--------------------------------------
```

```
result:     00000000000000001010010000  (11th, 7th, and 4th bits are set)
```

After processing 'l', checker = 00000000000000001010010000.

Character 4: 'o'

- 'o' has an ASCII value of 111.
- The position val is:
  val = 'o'.charCodeAt(0) - 'a'.charCodeAt(0) = 111 - 97 = 14.
  So, val = 14, meaning 'o' corresponds to the 14th bit.
- **1 << val**:
  1 << 14 shifts 1 to the 14th position, so we get:
  00000000000010000000000000.
- **Update checker**:
  Now, checker |= (1 << val) updates the checker to set the 14th bit:

sql

Copy code

```
checker:    00000000000000001010010000  (from 'h', 'e', 'l')

1 << 14:    00000000000010000000000000

---------------------------------------

result:     00000000000010001010010000  (14th, 11th, 7th, and 4th bits are set)
```

After processing 'o', checker = 00000000000010001010010000.

Character 5: 'l' (again)

- We already processed 'l' before, so the position val is still 11.
- **1 << val**:
  1 << 11 shifts 1 to the 11th position, so we get:
  00000000000000001000000000 (same as before).
- **Check if 'l' has been seen**:
  Now, when we perform checker & (1 << val), we get:

sql

Copy code

```
checker:    00000000000010001010010000  (current value)

1 << 11:    00000000000000001000000000

----------------------------------------

result:     00000000000000001000000000  (non-zero result)
```

Since the result is **non-zero**, it means the bit for 'l' is already set, indicating that we have seen 'l' before. This is a duplicate character.

Thus, the function would return false for the string "helol" because there is a repeated 'l'.

Summary:

- For the string "helol", the process detects a duplicate 'l' by checking if its corresponding bit is already set in checker. When it encounters the second 'l', the bit for 'l' is already set, and it returns false.