Solution 21: Python Set Remove Duplicates

1) `list(set(...))` (what does it print?)

One possible output (order may vary):

```
['C', 'B', 'A']
3
```

2) Numbers with duplicates (what does it print?)

One possible output (order may vary in the first line):

```
[3, 2, 1]
True
```

3) Remove duplicates (write code)

One correct answer:

```
items = ["pen", "pencil", "pen", "marker", "pencil"]

unique = list(set(items))
print(len(unique))
```

(The order of `unique` is not important here.)

4) Fill in the blanks (stable order)

```
numbers = ["A", "B", "B", "C", "A"]

unique = []
seen = set()  # create an empty set

for x in numbers:
    if x not in seen:
        unique.append(x)
        seen.add(x)

print(unique)
```

5) Stable order output (what does it print?)

```
[2, 3, 1]
3
```

6) Fix the bug (stable order)

Corrected code (one correct answer):

```
animals = ["cat", "dog", "cat", "bird", "dog"]

unique = []
seen = set()

for x in animals:
    if x not in seen:
        unique.append(x)
        seen.add(x)  # fix: update the set

print(unique)
```

7) Trace it (fill in the table)

Step	x	unique (list)	seen (set)
0 (start)	—	`[]`	`set()`
1	5	`[5]`	`{5}`
2	5	`[5]`	`{5}`
3	2	`[5, 2]`	`{5, 2}`
4	5	`[5, 2]`	`{5, 2}`
5	2	`[5, 2]`	`{5, 2}`
6	9	`[5, 2, 9]`	`{5, 2, 9}`

(Inside a set, the order does not matter.)

8) Stable order (write code)

One correct answer:

```
words = ["hi", "hi", "bye", "hi", "yes", "bye"]

unique = []
seen = set()

for x in words:
    if x not in seen:
        unique.append(x)
        seen.add(x)

print(unique)
```

9) Choose the best answer (multiple choice)

B

10) Time complexity (fill in the blanks)

10) Time complexity (fill in the blanks)

- `x in some_list` is **O(n)** (average).
- `x in some_set` is **O(1)** (average).

11) Count unique items (what does it print?)

```
4
True
False
```

12) Find the duplicates (write code)

One correct answer:

```
items = ["A", "B", "A", "C", "B", "B"]

seen = set()
dupes = set()

for x in items:
    if x in seen:
        dupes.add(x)
    else:
        seen.add(x)

print(dupes)
```

`dupes` is `{'A', 'B'}` (order may vary).

13) Remove duplicates from two lists (write code)

One correct answer:

```
a = ["Alice", "Bob", "Bob", "Chelsea"]
b = ["Bob", "David", "Alice", "Eva"]

unique_names = []
seen = set()

for x in a:
    if x not in seen:
        unique_names.append(x)
        seen.add(x)

for x in b:
    if x not in seen:
        unique_names.append(x)
        seen.add(x)

print(unique_names)
```

---

14) Keep the *last* time we see each item (stable last)

Fill-ins (one correct way):

```
nums = [1, 2, 1, 3, 2]

unique_rev = []
seen = set()

for x in nums[::-1]:   # go from right to left
    if x not in seen:
        unique_rev.append(x)
        seen.add(x)

unique = unique_rev[::-1]     # reverse unique_rev
print(unique)
```

---

15) Quick duplicate check (fill in the blanks)

```
l = ["x", "y", "z"]

s = set(l)

if len(s) == len(l):
    print("no duplicates")
else:
    print("duplicates")
```