String and Set-Based Python Programming Solutions

Set-Based Real-Life Python Solutions (Without Functions)

1. Remove duplicate email addresses from a mailing list.

```
email_list = ['a@example.com', 'b@example.com', 'a@example.com']

unique_emails = list(set(email_list))

print(unique_emails)
```

2. Track unique website visitors using set of IPs.

```
ip_list = ['192.168.1.1', '192.168.1.2', '192.168.1.1']

unique_ips = set(ip_list)

print(unique_ips)
```

3. Find common users between two product sales using sets.

```
users_product1 = {'Alice', 'Bob', 'Charlie'}

users_product2 = {'Bob', 'David'}

common = users_product1 & users_product2

print(common)
```

4. Find items common in two festival shopping lists.

```
list1 = ['sweets', 'crackers', 'lights']

list2 = ['lights', 'decorations']

common_items = list(set(list1) & set(list2))

print(common_items)
```

5. Check if a winning ticket number is in the participant set.

```
participants = {1001, 1002, 1003}
```

```
ticket_number = 1002

print(ticket_number in participants)
```

6. Check which candidate skills match job requirements.

```
candidate_skills = {'Python', 'SQL', 'Excel'}

job_skills = {'Python', 'Java'}

matched = candidate_skills & job_skills

print(matched)
```

7. Find common symptoms among multiple patients.

```
p1 = {'fever', 'cough'}

p2 = {'fever', 'headache'}

p3 = {'fever', 'nausea'}

common = p1 & p2 & p3

print(common)
```

8. Suggest mutual friends using intersection of sets.

```
user1_friends = {'Sam', 'John', 'Amy'}

user2_friends = {'John', 'Alice'}

mutual = user1_friends & user2_friends

print(mutual)
```

9. Check which products are missing from stock.

```
required = {'pen', 'notebook', 'eraser'}

available = {'pen', 'eraser'}
```

```
missing = required - available

print(missing)
```

10. Find who responded to both or only one of two surveys.

```
survey1 = {'Tom', 'Jerry', 'Spike'}

survey2 = {'Jerry', 'Tyke'}

both = survey1 & survey2

only_s1 = survey1 - survey2

only_s2 = survey2 - survey1

print('Both:', both)

print('Only Survey 1:', only_s1)

print('Only Survey 2:', only_s2)
```

11. Check if a directory contains duplicate files.

```
files = ['doc1.txt', 'doc2.txt', 'doc1.txt']

has_duplicates = len(files) != len(set(files))

print('Duplicates exist:', has_duplicates)
```

12. Get users interested in both music and sports.

```
music = {'Alex', 'Brian', 'Cathy'}

sports = {'Brian', 'David'}

both = music & sports

print(both)
```