

List-Based Real-Life Python Problems with Solutions

Problem: Add/remove items from a shopping cart and calculate total.

```
cart = [("Pen", 10), ("Book", 50)]
cart.append(("Pencil", 5))
cart.remove(("Pen", 10))
total = sum(item[1] for item in cart)
print("Total:", total)
```

Problem: Store book names and remove a book once it's issued.

```
books = ["Book1", "Book2", "Book3"]
issued = "Book2"
books.remove(issued)
print("Available books:", books)
```

Problem: Maintain a to-do list and mark completed tasks.

```
todo = ["Task1", "Task2", "Task3"]
completed = "Task1"
todo.remove(completed)
print("Remaining tasks:", todo)
```

Problem: Count votes from a list and find the winner.

```
votes = ["A", "B", "A", "C", "A"]
winner = max(set(votes), key=votes.count)
print("Winner:", winner)
```

Problem: Get average, max, and min of student marks.

```
marks = [70, 85, 90, 60]
print("Average:", sum(marks)/len(marks))
print("Max:", max(marks))
print("Min:", min(marks))
```

Problem: Maintain list of team members, add/remove dynamically.

```
team = ["Alice", "Bob"]
team.append("Charlie")
team.remove("Alice")
print("Current team:", team)
```

Problem: Track stops from origin to destination using a list.

```
stops = ["Start", "Mid", "End"]
for stop in stops:
```

```
print("Stop:", stop)
```

Problem: Store and plot stock price trends over a week.

```
import matplotlib.pyplot as plt
prices = [100, 102, 105, 107, 110]
days = ["Mon", "Tue", "Wed", "Thu", "Fri"]
plt.plot(days, prices)
plt.title("Stock Prices Over a Week")
plt.xlabel("Days")
plt.ylabel("Prices")
plt.grid()
plt.show()
```

Problem: Sort daily sales to find highest earning day.

```
sales = [("Mon", 100), ("Tue", 250), ("Wed", 180)]
highest_day = max(sales, key=lambda x: x[1])
print("Highest earning day:", highest_day)
```

Problem: Merge two user watchlists and remove duplicates.

```
watch1 = ["Movie1", "Movie2"]
watch2 = ["Movie2", "Movie3"]
merged = list(set(watch1 + watch2))
print("Merged watchlist:", merged)
```

Problem: List of students present, find absentees.

```
all_students = {"John", "Mike", "Sara"}
present = {"John", "Sara"}
absentees = all_students - present
print("Absentees:", absentees)
```

Problem: Store feedback messages and count negative ones.

```
feedbacks = ["Good", "Bad", "Poor", "Excellent", "Bad"]
negatives = [f for f in feedbacks if f in ["Bad", "Poor"]]
print("Negative feedback count:", len(negatives))
```

Problem: Store AQI values and classify each day.

```
aqi_values = [50, 120, 250]
for aqi in aqi_values:
    if aqi <= 100:
        print("Good")
    elif aqi <= 200:
        print("Moderate")
    else:
```

```
print("Poor")
```