1. What does `FILTER(Sales, Sales[Amount] > 1000)` return?

It returns a **table** containing only the rows from `Sales` where the column `Amount` is greater than 1000.
It does not aggregate—just filters rows.

---

2. Write a measure High Sales that sums Amount where Amount > 1000 using FILTER.

```
High Sales =
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(Sales, Sales[Amount] > 1000)
)
```

This calculates total sales only for transactions above 1000.

---

3. How does `ALLEXCEPT(Sales, Sales[Region])` differ from `ALL(Sales)`?

- `ALL(Sales)` removes **all filters** from the Sales table.
- `ALLEXCEPT(Sales, Sales[Region])` removes all filters **except Region** (so Region filtering stays).

---

4. Use SWITCH to categorize Amount:

```
Amount Category =
SWITCH(
    TRUE(),
    Sales[Amount] > 1000, "High",
    Sales[Amount] >= 500 && Sales[Amount] <= 1000, "Medium",
    "Low"
)
```

---

5. What is the purpose of ALLSELECTED?

`ALLSELECTED` respects slicers (user selections) but ignores row/column context from visuals.
Example: In a pivot table, it lets you calculate percentages relative to what the user selected, not the entire dataset.

---

6. Write a measure Regional Sales % showing each sale’s contribution to its region’s total (use ALLEXCEPT).

```
Regional Sales % =
DIVIDE(
    SUM(Sales[Amount]),
    CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Region]))
)
```

---

7. Create a dynamic measure using SWITCH to toggle between SUM, AVERAGE, and COUNT of Amount.

```
Dynamic Amount =
SWITCH(
    SELECTEDVALUE(Metrics[Choice]),   -- assume Metrics table has "SUM",
"AVERAGE", "COUNT"
    "SUM", SUM(Sales[Amount]),
    "AVERAGE", AVERAGE(Sales[Amount]),
    "COUNT", COUNT(Sales[Amount])
)
```

---

8. Use FILTER inside CALCULATE to exclude "Furniture" sales (`Products[Category] = "Furniture"`).

```
Non-Furniture Sales =
CALCULATE(
    SUM(Sales[Amount]),
    FILTER(Products, Products[Category] <> "Furniture")
)
```

---

9. Why might ALLSELECTED behave unexpectedly in a pivot table?

Because `ALLSELECTED` depends on **slicer + visual context**. If the pivot table has nested row/column fields, the calculation can behave differently depending on the layout.

---

10. Write a measure that calculates total sales and ignores filters from region.

```
Total Sales Ignore Region =
CALCULATE(
    SUM(Sales[Amount]),
    ALL(Sales[Region])
)
```

---

11. Optimize this measure:

```
-- Original:
High Sales = CALCULATE(SUM(Sales[Amount]), FILTER(Sales, Sales[Amount] >
1000))

-- Optimized:
High Sales =
CALCULATE(
    SUM(Sales[Amount]),
    Sales[Amount] > 1000
)
```

Using a Boolean filter inside CALCULATE is faster than wrapping FILTER.

12. Write a measure Top 2 Products using TOPN and FILTER to show the highest-grossing products.

```
Top 2 Products =
CALCULATE(
    SUM(Sales[Amount]),
    TOPN(2, VALUES(Products[ProductName]), SUM(Sales[Amount]), DESC)
)
```

13. Use ALLSELECTED with no parameters to respect slicers but ignore visual-level filters.

```
Sales Respect Slicers =
CALCULATE(
    SUM(Sales[Amount]),
    ALLSELECTED()
)
```

14. Debug: A SWITCH measure returns incorrect values when fields are added to a matrix visual.

Likely because `SELECTEDVALUE()` returns **blank** when multiple values exist in context.
Fix: Wrap it with `COALESCE` or use aggregation.

15. Simulate a "reset filters" button using ALL in a measure.

```
Reset Sales =
CALCULATE(
    SUM(Sales[Amount]),
    ALL(Sales)
)
```

This ignores all applied filters—like pressing “reset”.