SRT technical questions

Notes
Please don’t spend much more than 15-20 mins or so on each question. You should put your code and any explanations in the answer.

Python:

PY-01. Unified Pricing Adapter
Your desk owns multiple pricing libraries, each with a slightly different signature:

```
# Library 1
def price_swap(notional, fixed_rate, start, end, curve):
   …

# Library 2
def swaption_pv(trade_dict, market_data):
   …

# Library 3
class CapFloorPricer:
   def price(self, cap, curve, vol_surface, **kwargs):
      …
```

Task

1. Design and implement a **single public interface** PricerAdapter.price(trade, market_state) -> float that detects the trade type and routes the call to the correct underlying pricer.
2. Ensure extensibility for new trade types without editing existing logic (hint: registry pattern or functools.singledispatch).
3. Provide pytest unit tests demonstrating:
   • Correct routing for IRS, swaption, and cap/floor examples
   • Graceful handling of an unsupported trade type (raises custom UnsupportedTradeError)

VBA:

VBA-1 : Trader-Tool Integrity Checker

Traders often copy-paste formulas, breaking precedents.
Write a VBA function ListBrokenFormulas(ws As Worksheet) As Collection that:

1. Scans every cell in ws for formulas.
2. Returns a Collection of Range objects whose formulas contain #REF!.
3. Provide a companion sub HighlightBrokenFormulas() that loops through the collection and shades each cell light-red with dark-red text.
4. Include a Debug.Print summary: total formulas scanned, total broken, percentage broken.

Add a brief comment on how you would integrate this into the workbook's Workbook_BeforeSave event to block saving if >2 % of formulas are broken.

Excel:

XL-1 : Cash-Flow Ladder with SEQUENCE() & LET()

A plain-vanilla swap pays semi-annually over 10 years.
Inputs
- Start date: C3
- Notional: C4
- Fixed rate: C5

Tasks (single dynamic-array formula):

1. Generate all payment dates (20 rows).
2. Compute year-fraction via ACT/360.
3. Output a 3-column spill: PaymentDate, YearFrac, FixedCashFlow.

Requirements:
- Use SEQUENCE() for date generation.
- Encapsulate variables with LET().
- Ensure the result auto-updates if C3 changes.