# **I have this function:**

“def apply\_dampening(base\_score: float, ingredients: list[str], STRONG\_NEG: set, MOD\_NEG: set, POSITIVE: set) -> float: # Normalize arguments ingredients = [normalize\_text(i) for i in ingredients] STRONG\_NEG = set(normalize\_text(i) for i in STRONG\_NEG) MOD\_NEG = set(normalize\_text(i) for i in MOD\_NEG) POSITIVE = set(normalize\_text(i) for i in POSITIVE)

# Flags for presence  
has\_strong\_neg = any(i in STRONG\_NEG for i in ingredients)  
has\_mod\_neg = any(i in MOD\_NEG for i in ingredients)  
has\_positive = any(i in POSITIVE for i in ingredients)  
  
# Apply dampening logic  
if has\_positive and has\_strong\_neg:  
 base\_score \*= 0.75 # Strong conflict  
elif has\_positive and has\_mod\_neg:  
 base\_score \*= 0.85 # Moderate conflict  
elif has\_positive:  
 base\_score \*= 1.05 # Slight boost if no negatives  
return round(base\_score, 2)”

Do I still need the one below?:

“def score\_item(item: str, ingredients: Optional[str]) -> Tuple[int, Dict[str, List[str]]]: base = 70 item\_text = normalize\_text(item) ing\_text = normalize\_text(ingredients or item) matched = {"positive": [], "moderate\_neg": [], "strong\_neg": [], "whole\_food": [], "ultra\_processed": []}

# Positive signals   
for kw in POSITIVE:   
 if kw in ing\_text:   
 matched["positive"].append(kw)   
   
# Negative signals (moderate)   
for kw in MOD\_NEG:   
 if kw in ing\_text:   
 matched["moderate\_neg"].append(kw)   
   
# Strong negatives   
for kw in STRONG\_NEG:   
 if kw in ing\_text:   
 matched["strong\_neg"].append(kw)   
   
# Item-level hints   
for kw in WHOLE\_FOOD\_HINTS:   
 if kw in item\_text:   
 matched["whole\_food"].append(kw)   
for kw in ULTRA\_PROCESSED\_HINTS:   
 if kw in item\_text:   
 matched["ultra\_processed"].append(kw)   
   
# Scoring logic (bounded)   
score = base   
score += 6 \* len(matched["positive"]) # per unique positive   
score -= 7 \* len(matched["moderate\_neg"]) # per unique moderate negative   
score -= 15 \* len(matched["strong\_neg"]) # per unique strong negative   
score += 5 \* min(2, len(matched["whole\_food"])) # small boosts for whole-food hints   
score -= 10 \* min(2, len(matched["ultra\_processed"])) # small penalties for ultra-processed hints  
score = max(0, min(100, int(round(score))))   
return score, matched"