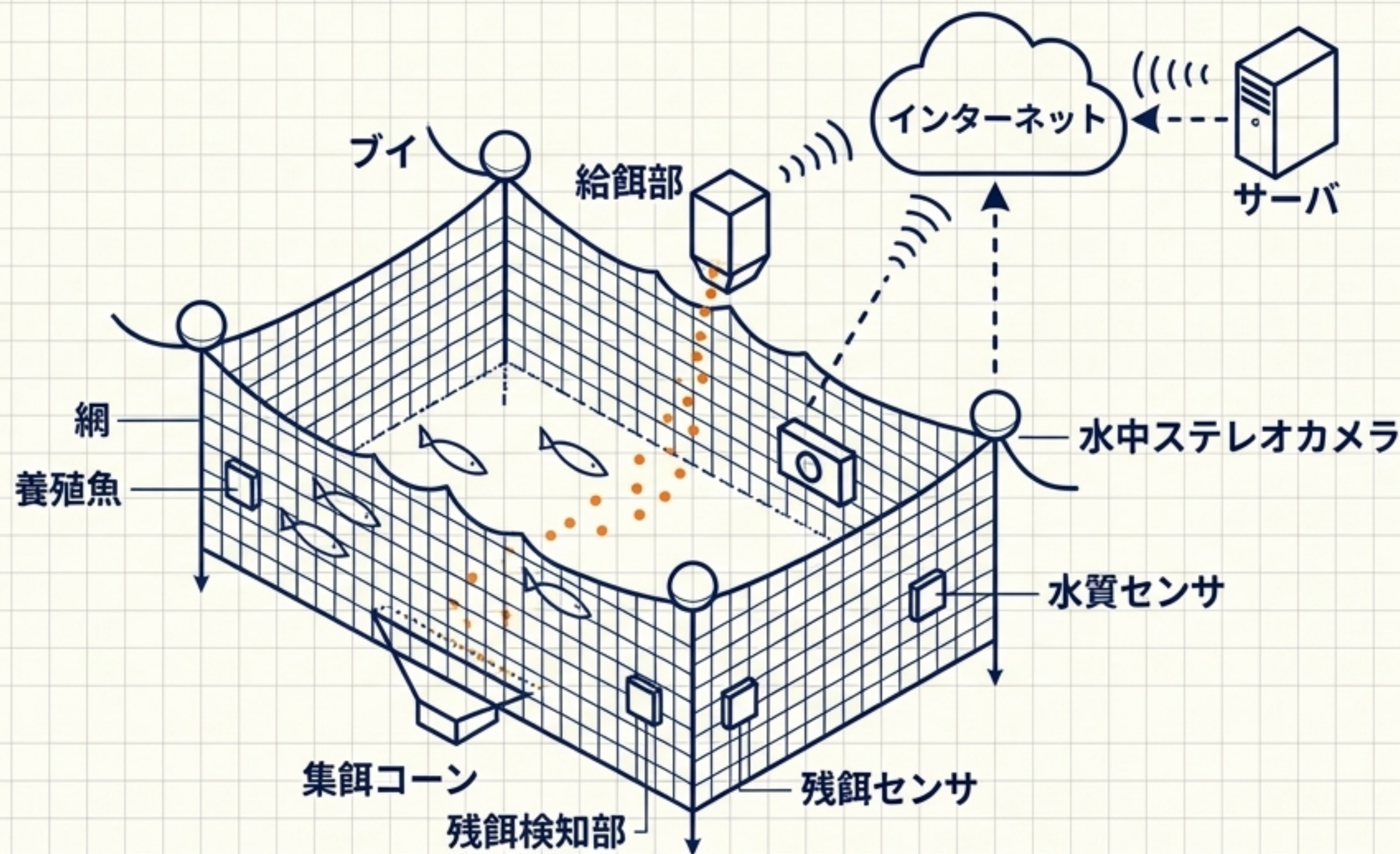


IoT生け簀システムの論理を解説せよ

応用情報技術者試験：設問3 完全攻略ガイド（思考の設計図）



MISSION BRIEFING: 本書の目的



単なる解答の提示
丸暗記のための解説



- システムを「設計者の視点」で読み解く思考訓練
- 物理現象とプログラムの繋がり（マッピング）の理解

あなたのミッション：散りばめられた手がかりを繋ぎ合わせ、自らの力で空欄 [a] ~ [f] を埋めること。

SYSTEM ARCHITECTURE: 誰が、何をしているのか？



給餌部

実行部隊。命令に従い
海へ餌を投下。



Server/AI

全体の頭脳。
「いつ」「どれだけ」
餌をやるか決定



Sensor (残餌検知部)

監視役。
食べ残して沈んできた
餌を「1個ずつ」数える。

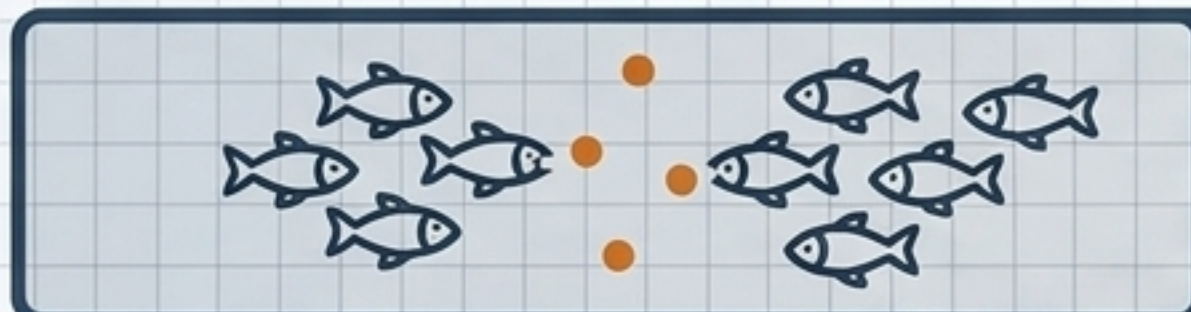
PHYSICAL TIMELINE: 生け簀の中で起きている物理現象

DEPTH (深さ)



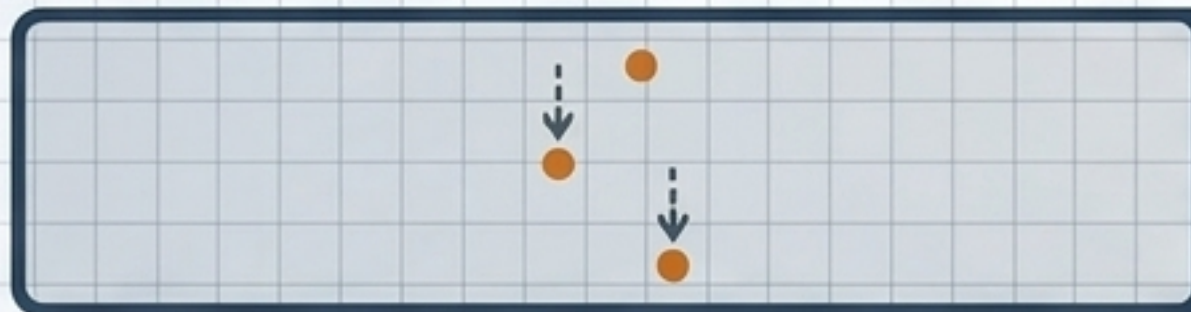
給餌 (Drop)

給餌部が指定された量の餌を海面に投下する。



捕食 (Eat)

養殖魚が群がり、餌を消費する。



沈下 (Sink)

食べ残された餌 (残餌) がゆっくりと底へ向かって沈む。



計測 (Count)

集餌コーンを通過した残餌を、センサーが1個ずつカウントする。



ヒント: 餌を投下してからセンサーが検知するまでには「タイムラグ (沈む時間)」が存在する。

THE TREASURE MAP: 設問3を解くための「鍵」の在り処

出典：令和3年度筑波大学附属高等学校入試問題第7

問7 IoTを利用した養殖システムに関する次の記述を読んで、問1～3に答えよ。

G社は、海上の生け簀の中で設置した養殖魚の育成を支援する、IoTを利用した養殖システム（以下、スマート生け簀という）を開発している。

〔スマート生け簀のシステム構成〕

スマート生け簀の概観を図1に、スマート生け簀のシス

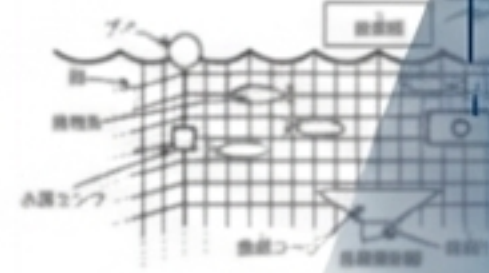


図1 スマート生け簀の概観



図2 スマート生け簀のシステム構成

スマート生け簀は、網をつるすためのブイに搭載される。餌投入装置は、餌投入制御装置と餌投入制御ユニットで構成される。餌投入制御装置は、餌投入制御ユニットで構成される。餌投入制御ユニットは、餌投入制御プログラム、餌投入制御ソフトウェア、餌投入制御ハードウェアで構成される。餌投入制御プログラムは、餌投入制御ソフトウェア、餌投入制御ハードウェアで構成される。餌投入制御ソフトウェアは、餌投入制御ハードウェアで構成される。餌投入制御ハードウェアは、餌投入制御システムで構成される。

〔給餌部の動作概要〕

- (1) 給餌部は、定期的に動画データ及び重量データを受信してフラッシュメモリに保存する。保存した動画データ及び重量データをサーバに送信する。
- (2) 給餌部は、給餌重量及び給餌日時をサーバから受信する。指示された給餌日時になると、給餌装置を起動する。

〔給餌部の動作概要〕

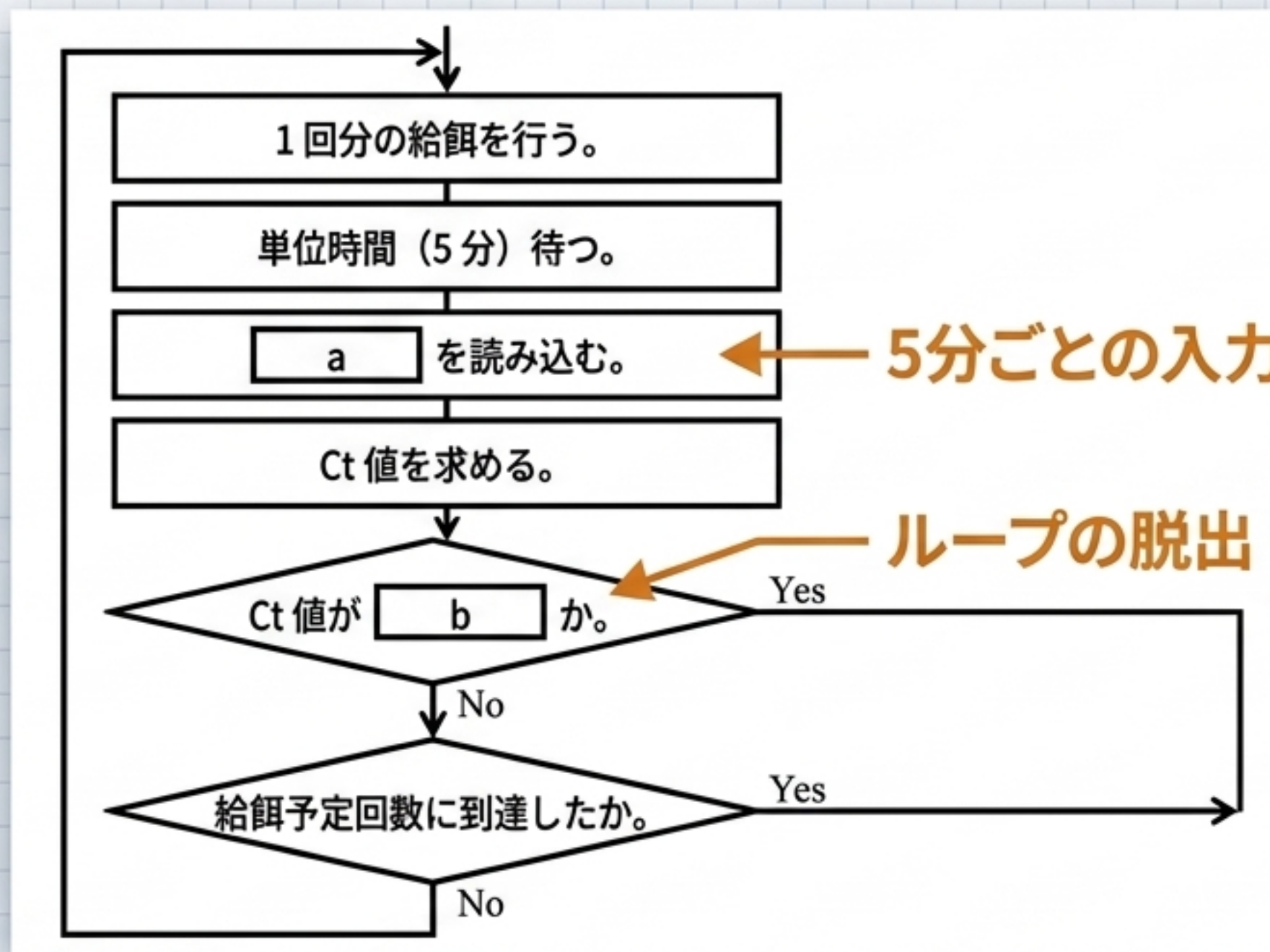
- (1) 給餌部は、定期的に動画データ及び重量データを受信してフラッシュメモリに保存する。保存した動画データ及び重量データをサーバに送信する。
- (2) 給餌部は、給餌重量及び給餌日時をサーバから受信する。指示された給餌日時になると、給餌装置を起動する。

〔給餌部の動作概要〕の(2)の後半部分

残餌センサは、通過する残餌を1個ずつカウントし、その値をを表示し込る。
給餌部は、5分間隔で[???]を読み込み、Ct値がしきい値以上のときは、
食べた餌の重量を求める。ここで、給餌を行う1個の餌の重量は均一とする。

解答へのすべてのロジックは、この数行に凝縮されている。
テキストの「言葉」とフローチャートの「図」を照らし合わせよ。

TARGET 1: フローチャートのループ構造を解剖する



これは単なる図形ではない。「5分間に1回、魚の食欲を確認する」というシステムの思考プロセスそのものである。

DECODING [a] : 「入力」を探せ

[a] を読み込む



Ct値を求める

フローチャートを見ると、[a] を読み込んだ直後に「Ct値を求める」処理を行っている。つまり、[a] はCt値を計算するための「材料」である。

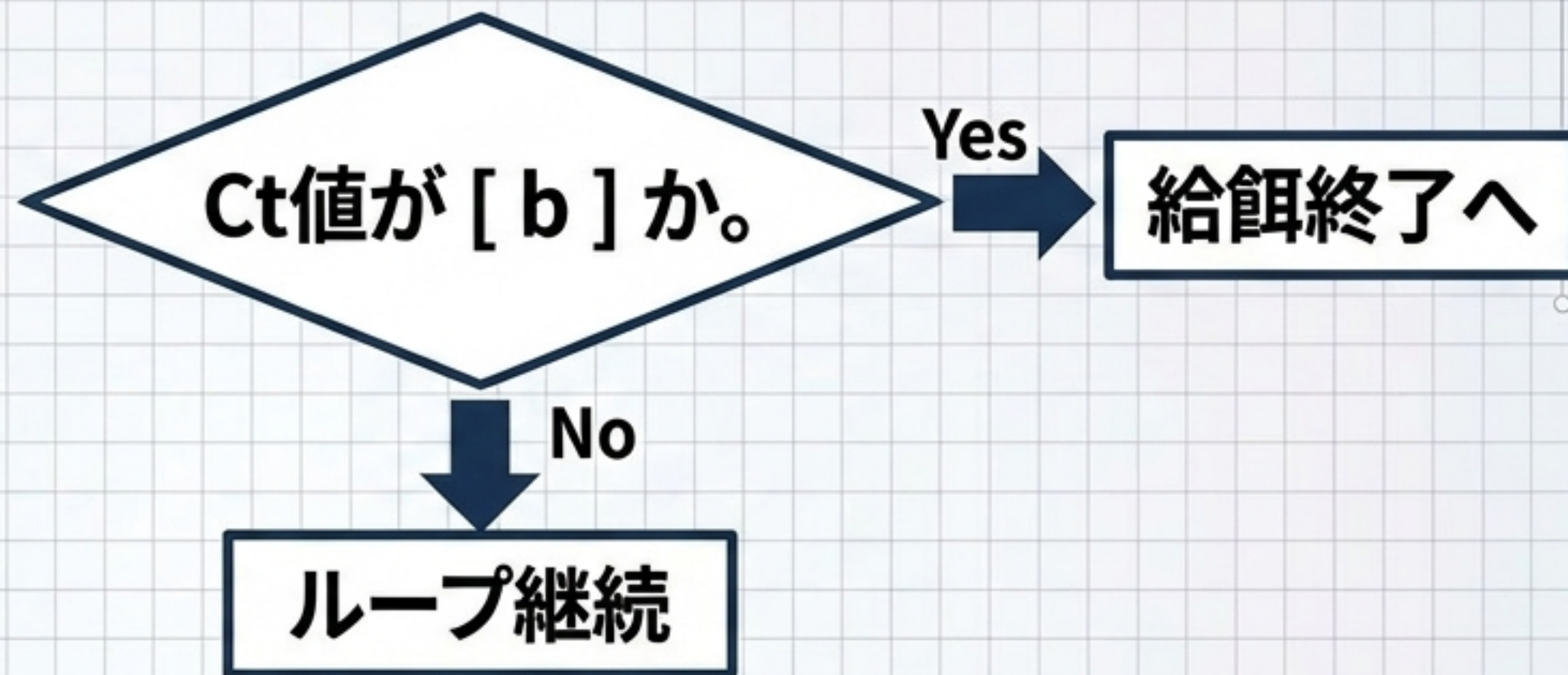
問題文のテキストを検索せよ。

給餌部は、5分間隔で【 ??? 】を読み込み、単位時間当たりのカウント値（以下、Ct値という）を求める。

物理世界のタイムラインを思い出そう。センサーは何を数えていたか？
その「値」を示す名詞が [a] に入る。

DECODING [b] : 「ループ脱出」の条件を探せ

1. 分岐の行き先に注目。
Yesに進むと「給餌終了」
ルートへ向かっている。
2. つまり、[b] は「システムが
給餌をやめる判断基準」
である。



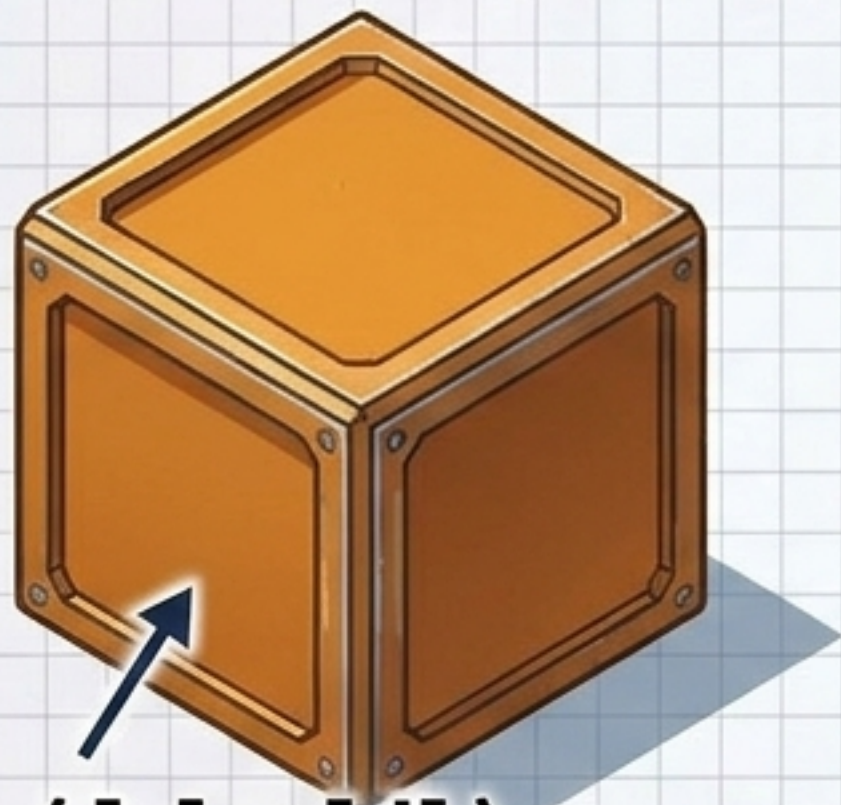
Ct値が【 ??? 】ときは、養殖魚が
餌を食べなくなると判断する。

【 ??? 】とき、給餌を終了する。

魚がお腹いっぱいになり、底に落ちてくる残餌のペース (Ct値) が
「ある基準」を超えたらストップする。その状態を指す言葉を探せ。

TARGET 2: 計算式を「視覚的」に理解する

$$\text{食べた餌の重量} = ([c] \times [d]) - ([e] \times [f])$$



$([c] \times [d]) =$

【投下した全体の重量】

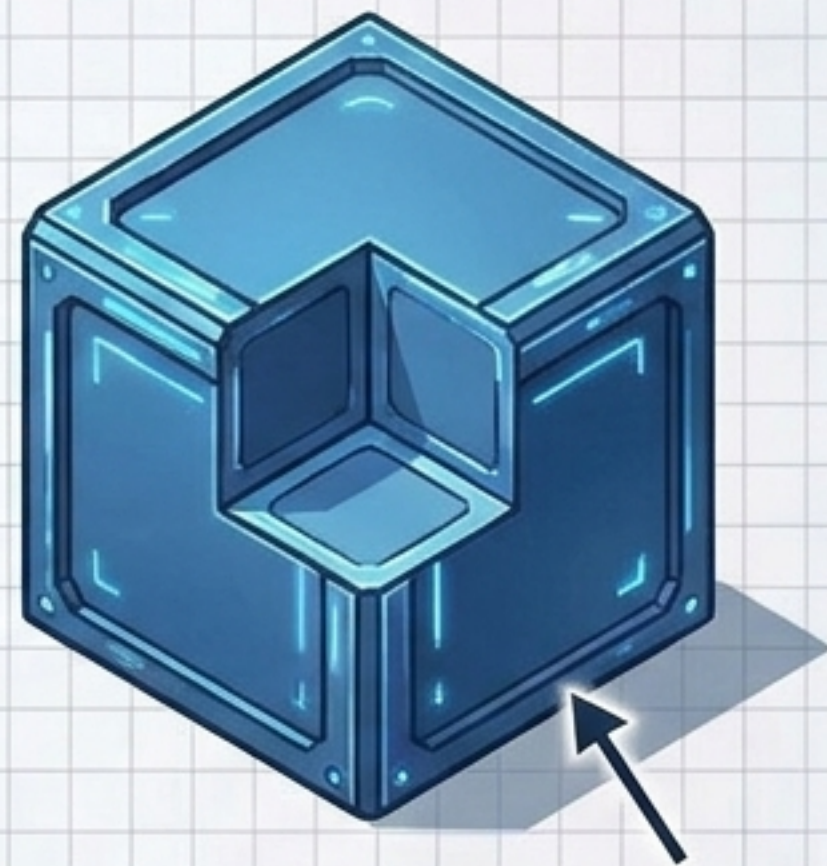
—



$([e] \times [f]) =$

【沈んできた残餌の重量】

=



【魚が実際に食べた量】

魚が食べた量は直接量れない。だから、「上から入れた量」から「下から出てきた量」を引く。極めてシンプルな引き算である。

FIRST TERM: 「投下した全体の重量」をどう導くか？

TARGET: ([c] × [d])



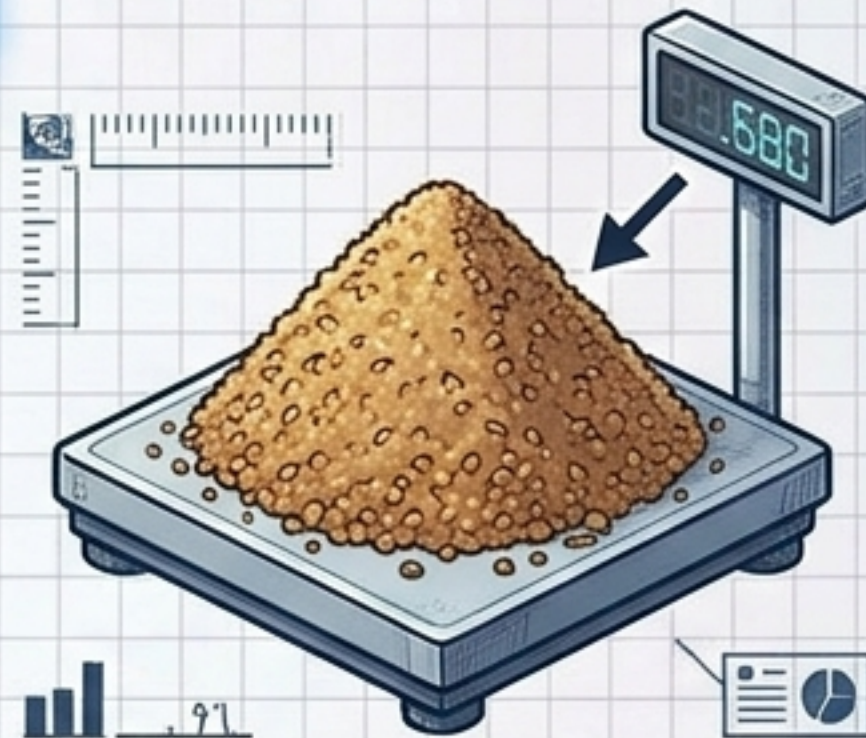
〔 1回分の重さ (g) 〕

×



〔 行った回数 (回) 〕

=



全体の重さ (g)

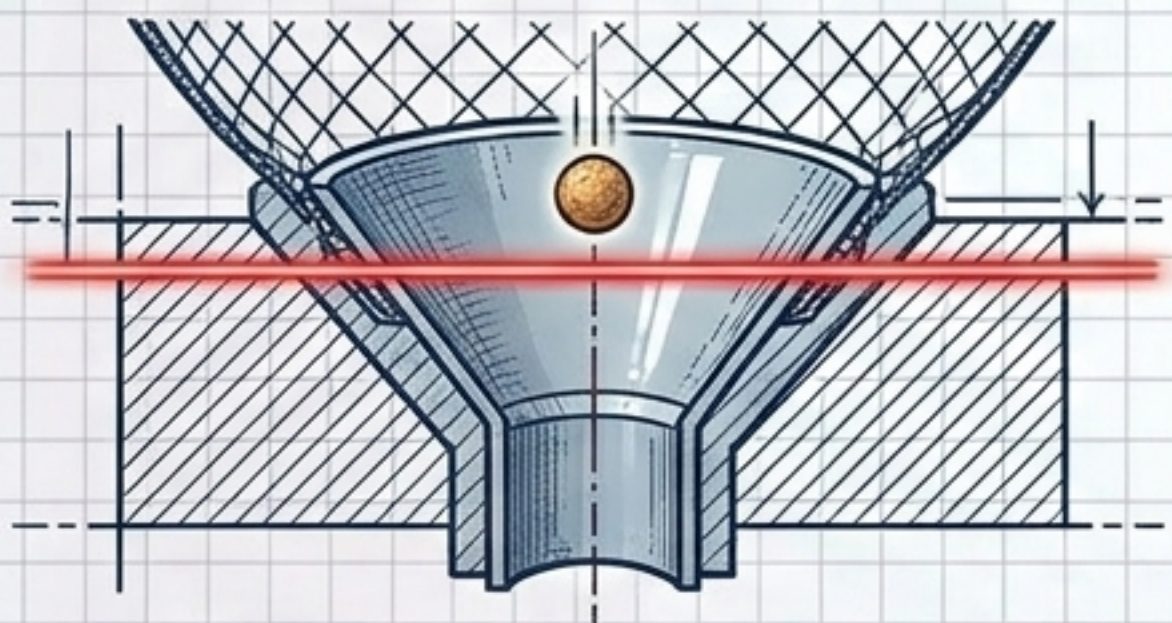
機械は「全体の重さ」を直接知ることにはできない。
知っているのは、「1回に落とす量」と、「落とした回数」だけである。

解答群の中から、「1回あたりの重量」を示す言葉と、「実際に投下した回数」を示す言葉のペアを見つけ出せ。(※「予定」と「実際」を間違えないこと!)

SECOND TERM: 「残餌の総重量」をどう導くか？

TARGET: ([e] × [f])

問題点: センサーは「重さ」を量れない。
通り抜けた粒を「1個、2個…」と数える
(カウントする) ことしかできない。



解決策: 個数を重さに変換する魔法の掛け算



The visual equation shows a single brown pellet on the left, followed by a large blue 'X' multiplication symbol, then a digital display showing '999' in blue, followed by an equals sign, and finally a pile of many brown pellets on the right.

[1個の重さ (g)] [カウントされた
総数 (個)] 残餌の重さ (g)

問題文の「給餌を行う1個の餌の重量は均一とする」という一文は、
この掛け算を成立させるための重要な伏線である。
解答群から「1個の重さ」と「最終的なカウント数」を探せ。

DIAGNOSTIC MATRIX: 解答群のカテゴリライズ

重量 (Weight / g)	個数・回数 (Count / n)	割合・その他 (Rate / Threshold)
イ: 1回当たりの餌の重量 ウ: 1個の餌の重量 カ: 給餌重量	キ: 給餌予定回数 ク: 給餌を行った回数 工: 給餌開始時の カウント値 オ: 給餌終了時の カウント値	ア: Ct値 (単位時間あたりの カウント)

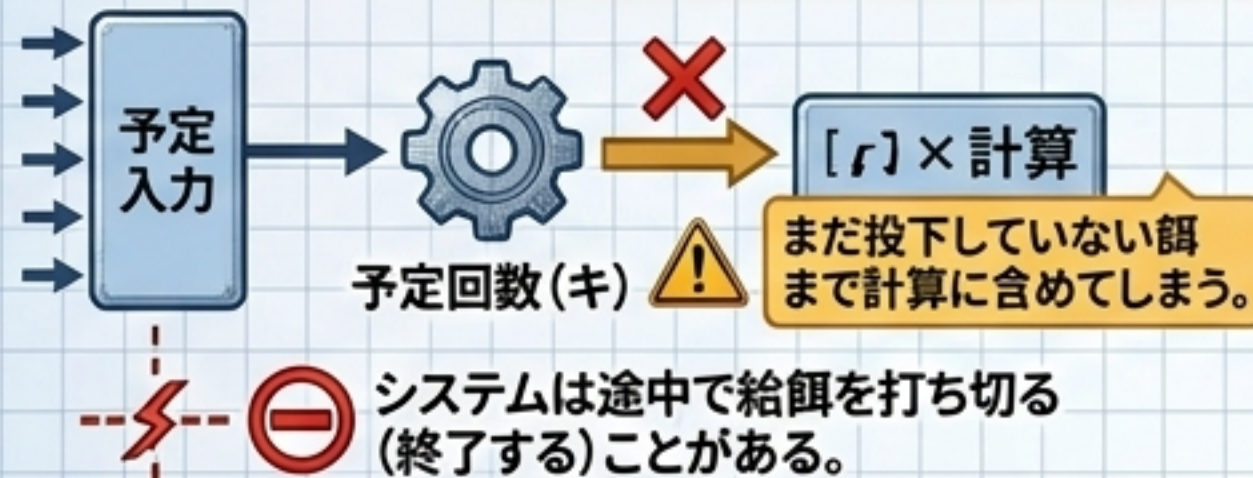
質量(g)を求める数式では、必ず「重量」と「回数・個数」を掛け合わせる必要がある。同じカテゴリ同士を掛けても答えは出ない。

SYSTEM WARNING: 陥りやすい2つの罠

WARNING

TRAP 1: 「予定」と「実績」の混同

予定
(PLANNED)



実績
(ACTUAL)



TRAP 2: 「Ct値」の誤用

Ct値(ア)は「5分間あたりのペース(速度)」である。



車のスピードメーターと同じ。



全体の距離(総重量)を出すためにスピードメーターの数値をそのまま足し引きしてはいけない。

速度 × 時間 = 距離

SYNTHESIS: フローと数式の接続

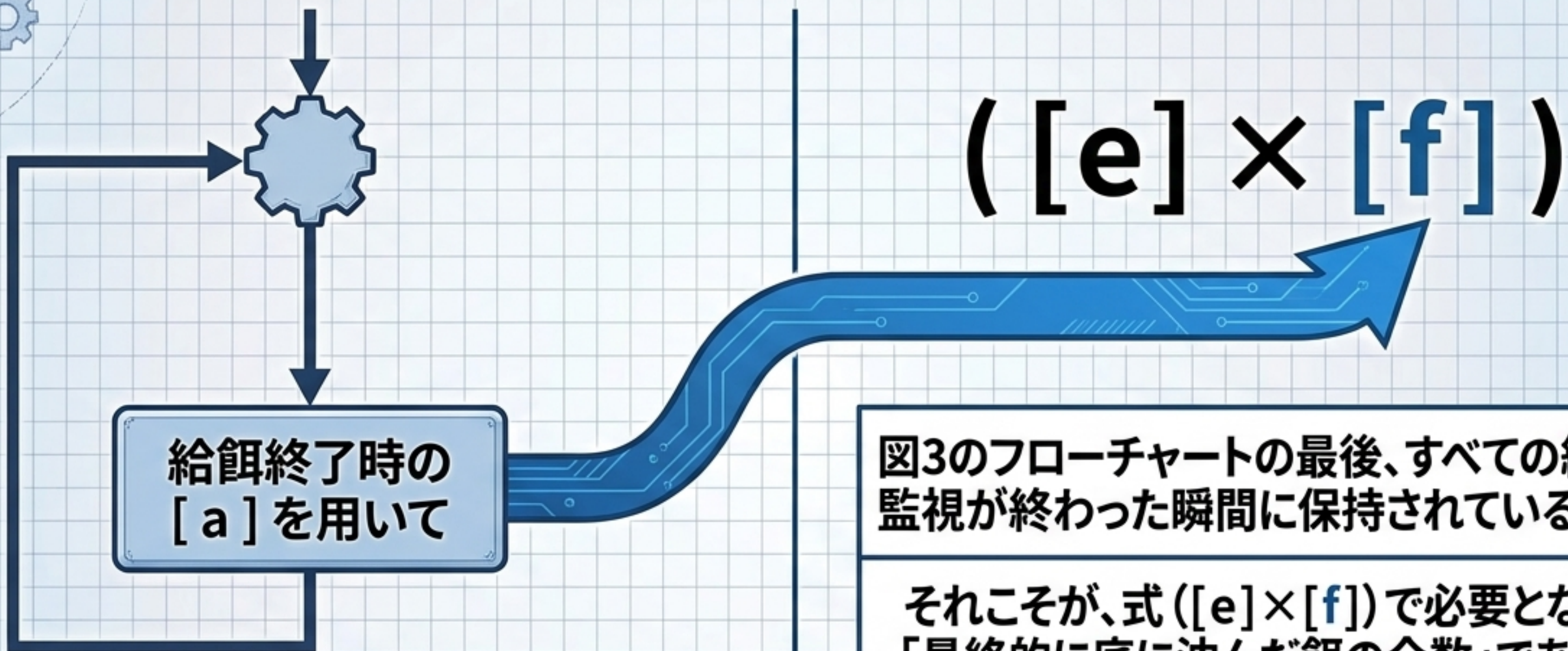


図3のフローチャートの最後、すべての給餌と監視が終わった瞬間に保持されている値。

それこそが、式 $([e] \times [f])$ で必要となる「最終的に底に沈んだ餌の全数」である。

物理世界でのアクション(投下と沈下)、ソフトウェアでのループ(監視と終了判定)、そして最終的な計算式(引き算)。これらすべては1本の線で繋がっている。

ALL PIECES ARE ON THE BOARD.

- ☐ フローの入力と脱出条件は特定したか？
- ☐ 全体の重量を求める掛け算のペアは見つけたか？
- ☐ 残餌の重量を求める掛け算のペアは見つけたか？
- ☐ 「罨」となる選択肢は除外したか？

[a]

[b]

[c]

[d]

[e]

[f]

思考の設計図は完成した。元の問題文に戻り、
自らの手で [a] から [f] の空欄を撃ち抜け。