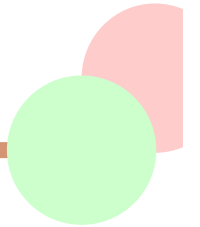


2014年度 認知情報処理論B
最終課題
- インド式掛け算 -



問題

◆ **インド式数学**を使って，2桁×2桁の掛け算を解く

日本式掛け算 $47 \times 2 + 47 \times 80$ を計算

$$\begin{array}{r} 47 \\ \times 82 \\ \hline 94 \\ 376 \\ \hline 3854 \end{array}$$

47 × 2 (green box)
47 × 8 (red box)

インド式掛け算 $7 \times 2 + 40 \times 80 + 40 \times 2 + 7 \times 80$ を計算

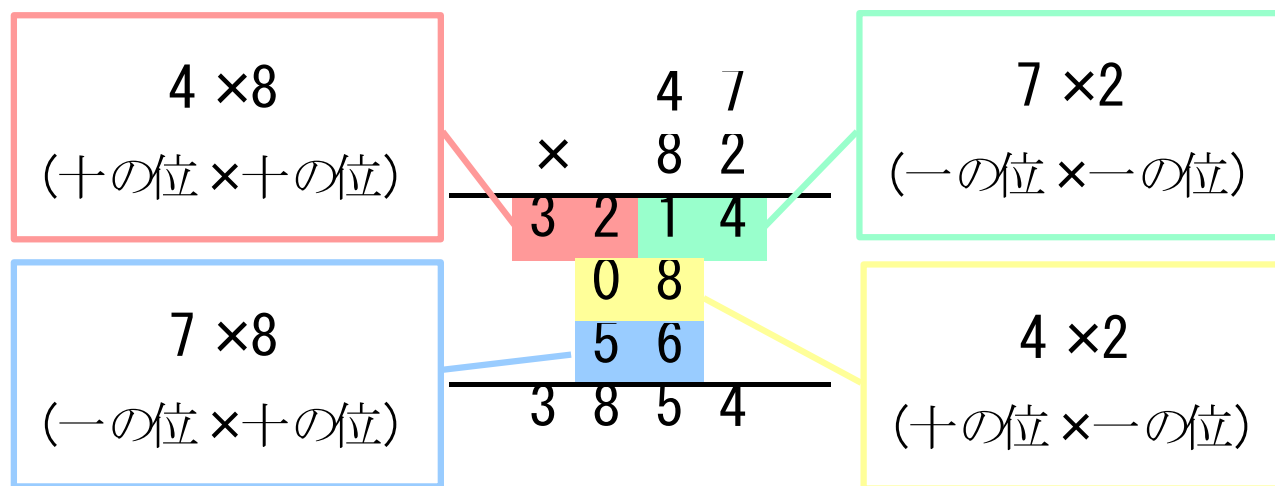
$$\begin{array}{r} 47 \\ \times 82 \\ \hline 3214 \\ 08 \\ 56 \\ \hline 3854 \end{array}$$

4 × 8 (十の位 × 十の位) (red box)
7 × 2 (一の位 × 一の位) (green box)
7 × 8 (一の位 × 十の位) (blue box)
4 × 2 (十の位 × 一の位) (yellow box)

インド式掛け算の基本ルール

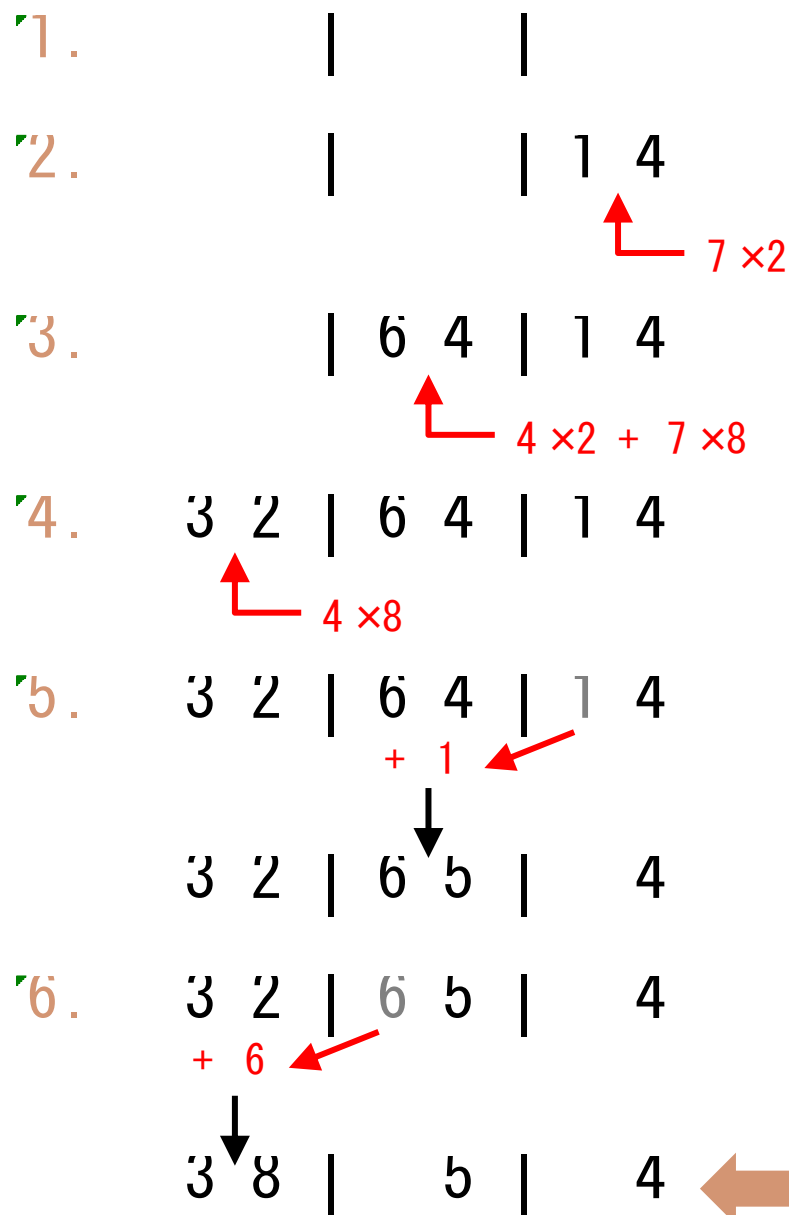
1. 全部の数の組み合わせの掛け算をする
2. 1の答えをそれぞれ位を間違えないように、桁を区切って書く
3. 2を足し算して答えを出す

例



➡ さらに簡単な計算方法として「横書き法」がある

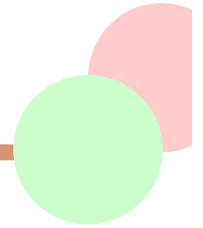
インド式掛け算の横書き法



1. 線を2本引き, 3ブロックに分ける
2. 右に「一の位×一の位」の答えを書く
3. 中央に「十の位×一の位」をたすき掛けで計算した答えを書く
4. 左に「十の位×十の位」の答えを書く
5. 右ブロックの数字の十の位を中央ブロックの数字に足す

出てきた数字を左から読むと答えとなる

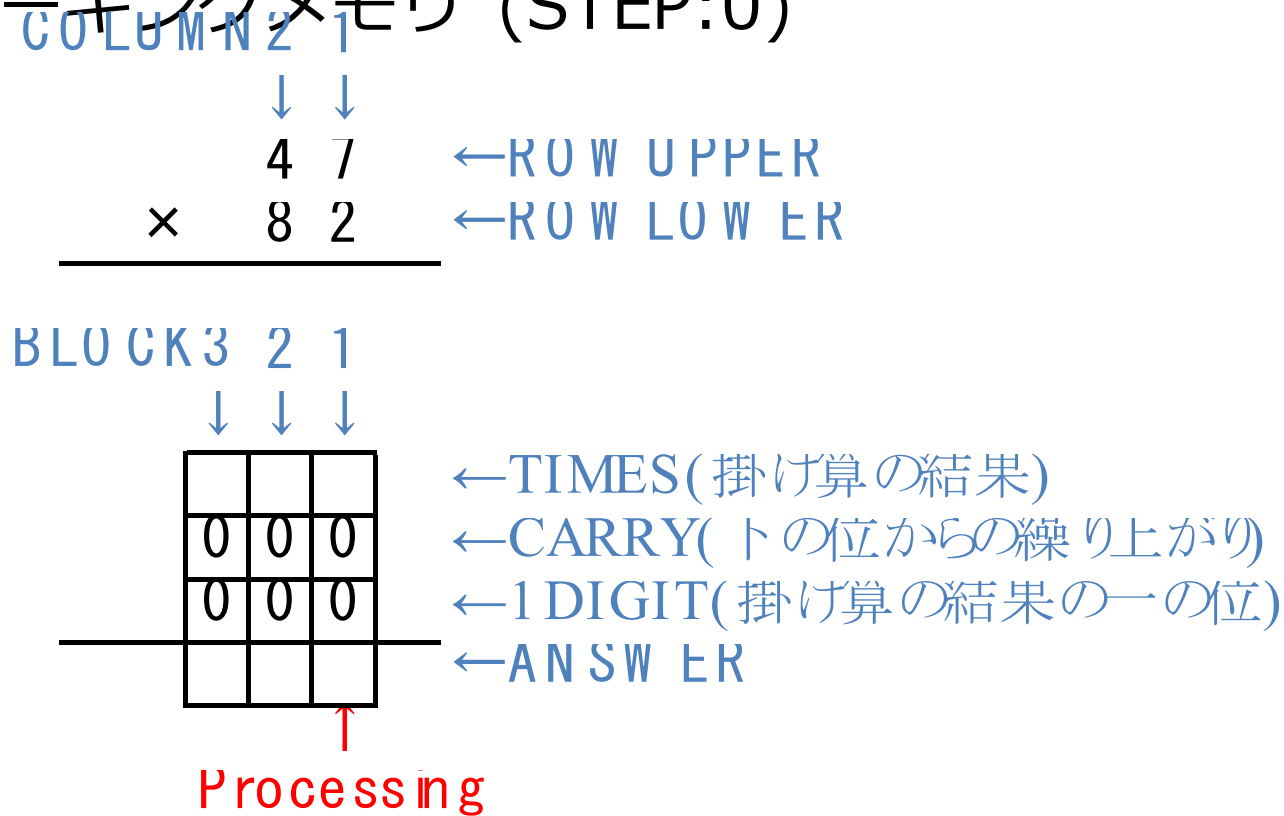
作成した認知モデル



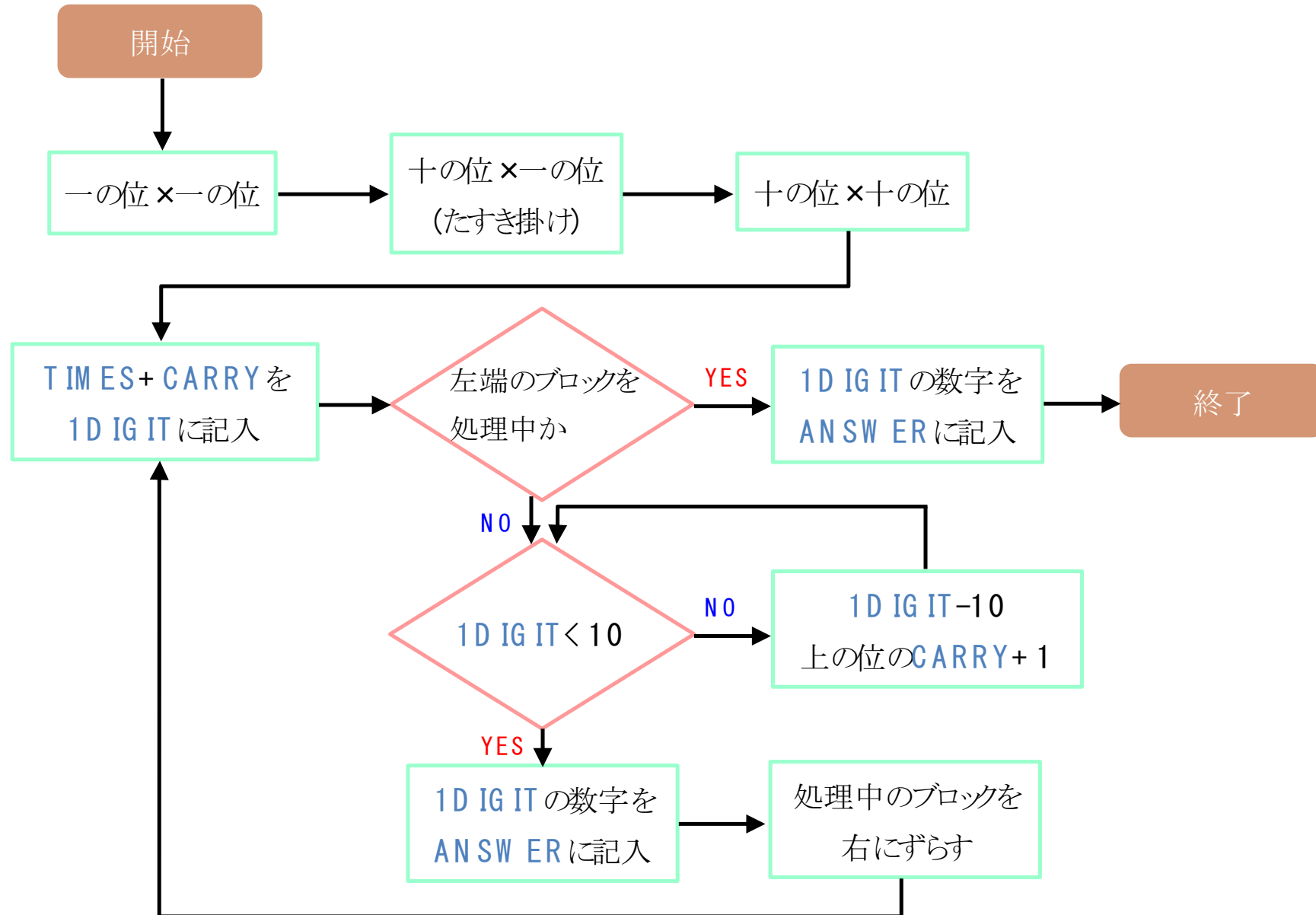
◆問題

2桁×2桁の掛け算をインド式掛け算の横書き法を使って解く

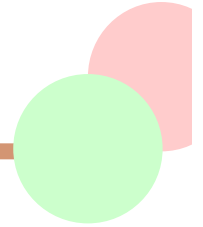
◆初期ワーキングメモリ (STEP:0)



状態遷移

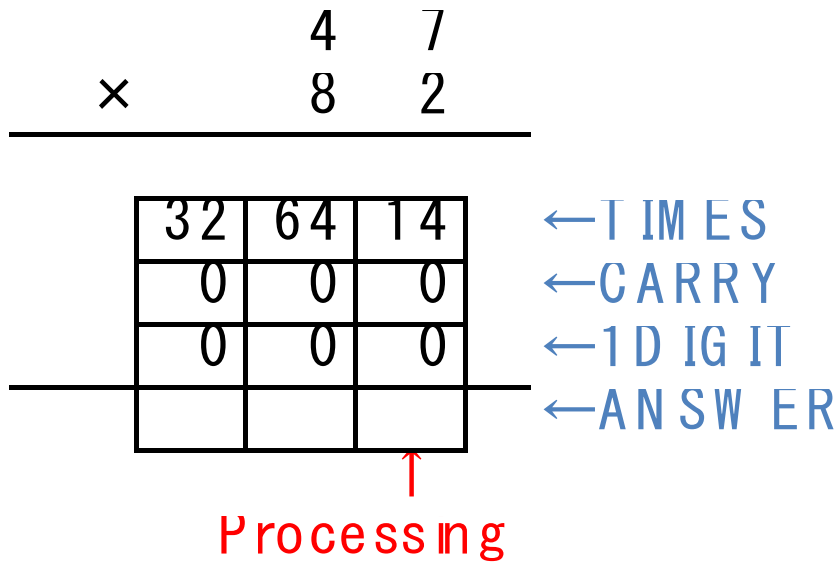


ワーキングメモリの推移



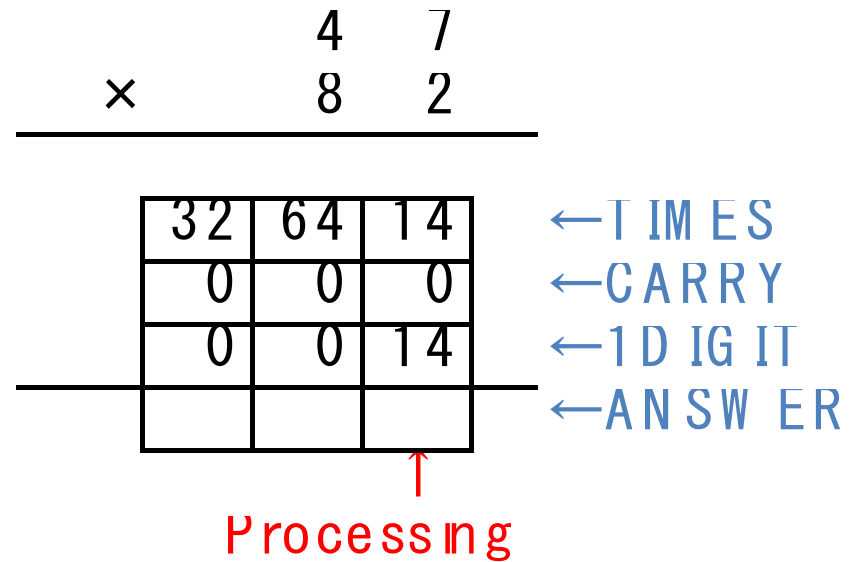
STEP:3

掛け算の結果をTIMESに記入

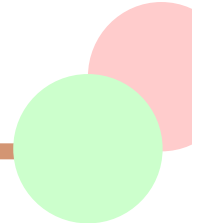


STEP:4

TIMES+CARRYを
1DIGITに記入

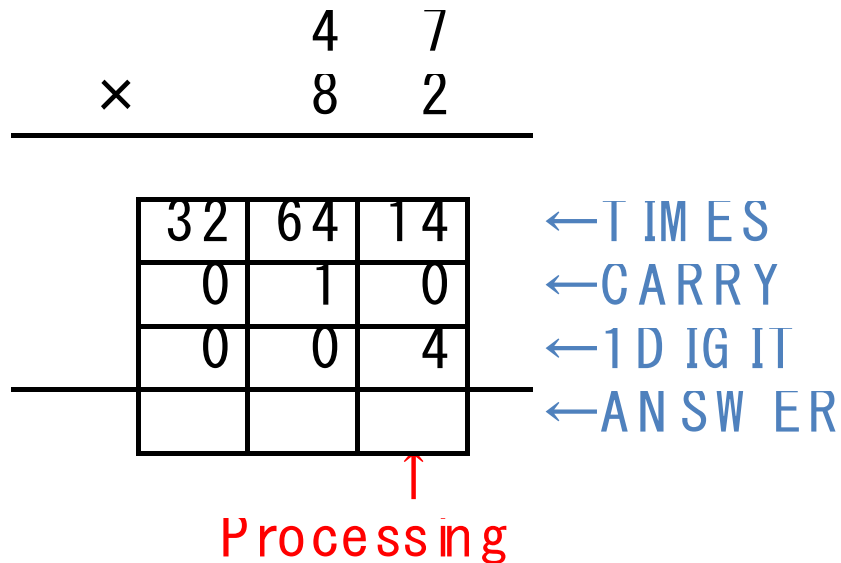


ワーキングメモリの推移



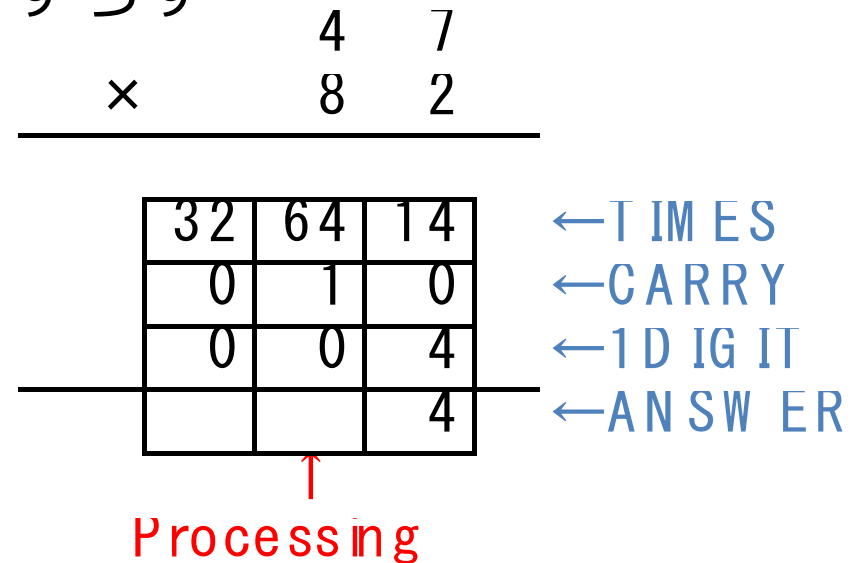
STEP:5

1DIGIT > 10 なので
1DIGITから10を引き、
上の位のCARRYに1を足す

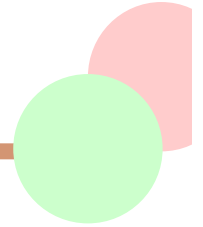


STEP:8

BLOCK1のANSWERが
記入されたので、
Processingを1ブロック左に
ずらす



ワーキングメモリの推移



STEP:38

Processingが左端にあるので
1DIGITをANSWERに記入

$$\begin{array}{r} \times \quad \quad 4 \quad 7 \\ \quad \quad \quad 8 \quad 2 \\ \hline \end{array}$$

32	64	14
6	1	0
38	5	4
38	5	4

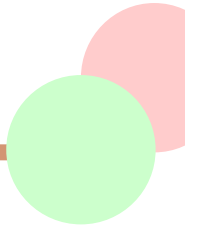
←TIMES
←CARRY
←1DIGIT
←ANSWER

↑
Processing

STEP:39

左端のANSWERが空でないの
で
処理を終了

終了までのステップ数比較



◆例1: 繰り上がりなし

$$\begin{array}{r} 1 \\ 1 \\ \hline \end{array}$$

1	0	0
0	0	0
1	0	0
1	0	0

ステップ数15で終了

◆例2: 繰り上がりあり

$$\begin{array}{r} 9 \\ 9 \\ \hline \end{array}$$

81	162	81
17	8	0
98	6	1
98	0	1

ステップ数40で終了

繰り上がりの数が最小・最大のものを実行した結果,
2桁×2桁の掛け算をインド式数学で解くと,
15~40ステップで終了することが分かった

まとめ



- ◆ 2桁×2桁の掛け算をインド式数学の横書き法で解くモデルを作成
最大40ステップで計算が完了
- ◆ 同様の手順で2桁×3桁, 3桁×3桁, …の計算もインド式数学で解くことができる (今回は未実装)