

## パズルの解法を通じて学ぶプログラミングの基礎

～ 分割統治法と構造化プログラミングの概念の理解～

研究代表者 山 上 通 恵

共同研究者 川 崎 百 合

富 田 篤

渡 部 知 也

田 向 尚 史

山之内 孝 至

石 橋 知 子

### 要約

情報教育の3つの柱とされる「情報活用の実践力」「情報の科学的な理解」「情報社会に参画する態度」は、普通教科「情報」の3つの科目「情報A」「情報B」「情報C」のいずれにおいても何らかの形で触れることになっており、各科目はそれぞれ、3つの柱のすべてを含みつつ、情報Aでは「情報活用の実践力」、情報Bでは「情報の科学的な理解」、情報Cでは「情報社会に参画する態度」に比重を置いた内容構成になっている。

ところで、平成20年度における全国の高等学校での教科書採択数の割合は、情報Aが73.4%、情報Bが10.7%、情報Cが15.9%である。一方、次期学習指導要領の改訂において、「情報の科学」「社会と情報」に科目構成が変わることが公表され、「情報の科学」は現「情報B」、「社会と情報」は現「情報C」であるとの認識が、正しいかどうかは別にして、教育現場に広がっている。また、そのような変更が実施された場合、発展的に解消されるといわれる「情報A」を実施している学校は、そのほとんどが「社会と情報」に流れる空気もある。もしこのような流れができてしまうと、圧倒的に多数の生徒が、「情報の科学的な理解」に対してより深い学習の機会を得られない状況ができてしまう。

本研究では、実際の教育現場では避けられることが多い「情報の科学的な理解」に焦点を当て、平易な題材、導入が容易な実践環境で研究を行い、指導者にも特別な知識や技術を求めることなく、学習の展開まで平易なレベルで完結することをねらい、普遍的な学習単元として確立することを目指した。

---

代表者勤務校：兵庫県立社高等学校  
(前任校：兵庫県立神戸甲北高等学校)

## 1. 本校の概要と本校における情報教育

本校は、1997年に普通科から総合学科に改組された県立高校で、1学年6クラス編成、全校生徒700人余りの学校である。普通科の同規模校に比べて、教員が10数人多く、1年次二人担任制を実施して、キャリアガイダンスや個別の「時間割づくり」に対応している。また進路実現をサポートし実力アップをはかるため、選択生徒数10人以下の講座が27講座、10人～27人の講座が107講座というような少人数の講座展開を実現している。

1年次の原則履修科目の「産業社会と人間」では、職業や進路について考えたり、学ぶ力や生きる力を身につけたりし、2年次の「総合的な学習の時間」では学び方を学び、自ら課題を見つけたり、自ら考えたりして研究を行うことを目指す。途中、分野別模擬講座(学問おもしろ体験)として、大学の先生を招聘して講義を開講し、興味・関心のある学問分野に参加したり、逆に大学へ出向いて本格的な施設での実習も体験したりして、3年次における卒業研究のテーマ探しのサポートにも役立てている。また、高大連携を推進し、専門分野のより高度な学習を通じて、学問・研究の楽しさを味わい、本校の卒業単位として認定されるようにもなった。3年次の「卒業研究」では、高校生活の集大成として、自ら課題を見つけ、論文や作品としてまとめ、生徒はこれらの成果を自らの進路に活かしている。

総合学科設立当初は、1年次に「産業社会と人間」、2年次に「情報に関する基礎的科目」、3年次に「課題研究」を、それぞれ原則履修科目として置いた。2003年までの数年間は、教科書もない、免許もない手探りの状態での実施であったが、他の総合学科が情報処理(商業科目)をなぞるような授業を展開している実態とは異なり、本校では、近い将来に新しい教科として「情報科」が立ち上がることを見越し、他の教科で代替できるようなものではない教科にしたいという共通理解をもった。結果として

- ・ 表現力を高める
- ・ コミュニケーション能力を高める
- ・ 資格取得を目指さない
- ・ ソフトウェアの操作指導に陥らない

などを目指すことを確認した。

2003年の現行の学習指導要領の実施に伴い、2年次に実施していた「情報に関する基礎的科目」は「情報C」として1年次に移し、高等学校の早い段階で情報活用技術を習得し、また表現力やコミュニケーション能力を高めることで、「産業社会と人間」や「課題研究」改め「卒業研究」、加えて他の教科の学習にも影響を及ぼす教科として位置付けた。他行に先駆けて実施した「情報に関する基礎的科目」の経験は、その後の「情報C」の展開に計り知れないアドバンテージを得た。

しかし、総合学科における実践として、3年次の「卒業研究」をまとめ上げることを強く意識した授業展開であったため、情報収集の手段や適切な扱い方、収集した情報の整理・活用方法、効果的な発表の仕方、知的財産権への配慮などの内容が続き、「情報の科学的な理解」は手薄であったことは否定できない。そ



写真1 少人数講座



写真2 課題研究集

ここで、2005 年から、

- ・ モニタは色をどのように表現しているか？
- ・ FAX はどのように情報を伝えているか？
- ・ 2進法を利用するメリットは？

といった軽い内容で、生徒の興味関心がどの程度のレベルなのか、またどのレベルの話まで理解できるのか、さらに担当者は自信を持って授業を展開できるのかといったことを試してみた。

いずれも、当然のことであるが、十分な準備と適切な教材によって、生徒の理解は期待以上のものがあり、以前の心配は杞憂であることがわかった。そこで、今回、学習指導要領では「言語の習得が目的とならないように」と釘を刺されている部分に、あえて立ち入る授業を試みた。

## 2. 研究主題と教材選定

研究の対象となる単元は、単純にプログラミングである。従来の「情報の科学的な理解」の指導においては、学習指導要領の記載通り、プログラミングにはほとんど立ち入らず、表計算ソフトの関数を使って合計や平均を求め、条件分岐関数で判断させる程度のものでしかない。その題材は生徒にとって教科書に掲載されているという必然性しかなく、強く興味をかきたてられるものではなかった。また、「言語の習得が目的とならないように」とあるが、ある程度の修得をしないと、「何ができるか分からない」というジレンマに陥り、教科書に書いてあること、教師の指示することをなぞるだけの実習になり、自ら考えで試行錯誤するということはできない。「特定の言語の習得を目的としない」ことを謳っているのかもしれないが、まず何か一つ言語を習得することで、言語だけではなく「プログラミングの考え方」も併せて習得でき、他の言語の習得の壁を取り払うことにもつながると

考えられる。

今回の研究においては、パズル(図1)という遊びの要素を感じさせる題材を使って、まず自分の頭で解を求める過程を体験し、その過程を書き出し、その書き出したものを定式化し、コーディングするという手順を踏ませる。この手順においても何度もフィードバックが発生し、複数の考え方の候補を思いつけること、複数の候補から一つを選び出す判断の根拠を考えること、などさまざまな寄り道があり、なかなか完成には至らないことも予想されるが、「どこかで見切りをつける」という判断を含め、問題解決の過程としてとらえることも体験させることができる。

こうした実践を1年間の「情報」の中に位置づけることによって、情報教育の3つの柱「情報活用の実践力」「情報の科学的な理解」「情報社会への参画」がよりバランスよくなり、次期学習指導要領の改定も見据えて、より科学的な領域へスムーズに移行できることを示したい。

題材として取り上げる数独というパズルは、その見た目において表計算ソフトウェアとの親和性が高いことがわかる。表計算ソフトの罫線も、パズルを表現するには十分なものである。

1	6			2			5	3
4	7		1		9		2	6
		8				1		
	4		3		2		6	
3				6				2
	9		7		5		4	
		4				7		
9	3		5		1		8	4
7	5	1		8			3	9

図1 数独

### 数独のルール

- ・ 各行に1から9までの数字を重複なく配置する。
- ・ 各列に1から9までの数字を重複なく配置する。
- ・ 各ブロック(太線で囲まれた3×3の領域)に1から9までの数字を重複なく配置する。

数独を表計算ソフトで解く過程を題材とすることは、ほかにも次のような利点がある。

- ・ 世界中で幼児から老人までが楽しんでいることから解るように、パズルのルールがきわめて単純であり、生徒の興味を持続させる要素を持っている
- ・ 大半の学校ですでにインストールされている表計算ソフト Microsoft Excel(以下 Excel)を利用することで、すべての生徒用コンピュータにプログラミング言語の学習をするためのソフトウェアを導入するコストが不要である
- ・ 自宅でも同じ環境を持つ生徒が多く、復習や発展的、自発的な学習を期待できる
- ・ 生徒の操作により「マクロの実行」を許可するだけで環境が整い、教師が事前に複雑な設定をする必要がない

### 3. 予備実践

#### (1) コンピュータ部の生徒を対象にした予備実践

1年目の2007年度は、コンピュータ部の生徒を対象に、予備実践を行った。部員の構成は表1のとおりである。

コンピュータ部の前身は数学研究部であったが、10年程前にコンピュータ部と名称を変更した。数学研究部時代は、純粋に数学を探究する部であったようであるが、コンピュータ部に姿を変えてからは、年によっては放課後にインターネットゲームを楽しむ部という時代もあった。

予備実践の対象とした部員は、3年生の男子はコンピュータの構造などにも興味を持つ生徒であるが、その他の部員は科学的な要素に興味はなく、コンピュータを使って絵を描いたり小説を作ったりするなど、デジタル文芸部といわれることもある。決して数学的、科学的素養を身につけた生徒ばかりではない集団である。

彼らとともに、文化祭が終わった6月下旬から、毎週木曜日の放課後のコンピュータ部の活動として次のような活動を行った。途中、定期考査や学校行事、担当者の出張など様々な要因で、立ち会う機会は予定よりも少なくなってしまうが、順調に実践が進んだ。このあと、より興味関心が薄いマルチメディア表現の選択者への予備実践、情報Cでの1年生全員への実践と展開されるが、すべての実践の基礎となるこの予備実践について詳述する。

表1 コンピュータ部部員構成

	男子	女子	計
1年次	0	1	1
2年次	2	1	3
3年次	2	0	2
計	4	2	6

#### 第01回(2007/06/21) ...オリエンテーション

生徒を集め、この研究の趣旨および今後の予定を説明した。生徒の発案で、毎回の活動の最初にパズルを1問解くことを決めた。この日は研究対象とするパズルの解説をして、実際にパズルを解かせた。6名の生徒のうち、このパズルを知っていたものは5名、そのうち実際に解いた経験のあるものは3名であった。またその3名全員が、ポータブルのゲーム機で提供されるもので解いた経験であり、必要に応じてヒントが提供されたり、途中誤りがないかをチェックしたりする機能などがあるとのことである。したがって、紙と鉛筆で最後まで解く過程は、全員が初体験であったと思われる。

#### 第02回(2007/06/28) ...思考過程の発見

第1回以降に、Webサイトや書籍で数独の別の問題を解いた生徒が4人いた。前回、決めた通り、手作業でパズルを解く。このとき、ひとつひとつの解を得た思考過程を記述させ、生徒間で比較させたところ、大まかに

解法1

一つの数字を決めて、その数字が入れない場所を消していき、決定できる場所を探す。(図2)

#### 解法2

個々のセルに入りうる数字を書きあげ、1つしか入り得ないセルを決定していく。(図3)

の2つの考え方に集約できた。

1	6			2			5	3
4	7		1		9		2	6
		8				1		
	4		3		2		6	
3				6				2
	9		7		5		4	
		4				7		
9	3		5		1		8	4
7	5	1		8			3	9

図2 解法1

1	6	9	48	2	478	489	5	3
4	7	35	1	35	9	8	2	6
25	2	8	46	3457	34678	1	79	7
58	4	157	3	19	2	589	6	1578
3	18	157	489	6	48	589	179	2
268	9	126	7	1	5	38	4	18
268	1268	4	269	39	36	7	1	15
9	3	26	5	7	1	26	8	4
7	5	126	246	8	46	26	3	9

図3 解法2

### 第03回(2007/07/05) ...思考過程の分析

前日に整理した2つの考え方について考察を深め、いずれか一方だけでパズルを解き、その特徴を考えた。図2の解法に比べ、図3の解法は決定できたセルがさかのぼって未決定のセルの候補に影響し思考が行き来すること、また見落としが多くなることがわかった。このことから、プログラミングの基礎の学習として、図1の解法を中心に考えることを決めた。

なお、このパズルを解く過程で、表計算ソフトに数値を入力し、解を探す生徒が現れたが、単に数値を入力するだけであり、セルに色をつけるなどの工夫はまだ見られなかった。

### 第04回(2007/07/12) ...Excelの機能

全員がExcelでパズルを解くことにした。単に数字を入力するだけでは、最初に与えられた数値との違いがわからなくなることが混乱を引き起こしたので、

- ・ 問題として与えられた数字のフォント色を赤にして目立たせるとともに、誤って変更できないようにセルをロックする

ことが生徒から提案され、その方法を指導した。

また、

- ・ その数字が入れないセルの背景を着色する

というアイデアが出され、その方法を指導した。いずれも、マウスを使って対象のセルを選択し、コマンドメニューからの操作で着色するというマニュアル操作であり、この時点ではプログラミングは行っていない。

### 第05回(2007/07/19) ...フローチャート

解法の流れを改めて文章で書き直し、その後フローチャートとして整理させた。ただし、「処理」と「分岐」のみの単純なものである。

夏休み前に具体的な VBA (Visual Basic for Applications) のコーディングの指導に入りたかったが、都合により夏休み明けに先送りした。思考の過程を文章化 (定式化) することで一つの節目としておいた。

## 第 06 回 (2007/09/06) ... 逐次処理とマルチステートメントおよび繰返し処理

この回から VBA のコーディングについて指導を始めた。まず Excel を起動し、配布したファイルを開かせ、VBE (Visual Basic Editor) の起動方法、Excel との切り替えを指導した。配布したファイルには、すでに数独の問題が入力されており、また罫線も引かれていて数独の盤面として完成されたものである。また、コマンドボタンもすでに 1 つ配置済みで、「実行」というキャプションも設定してある。これは、罫線の処理やボタン配置の操作の部分は今回の学習内容とは関係ないと判断し、すぐにコーディングにかかり自動処理を体験させたいと考えたからである。

さらに、パズル面の左上隅を A1 ではなく C3 に意図的に置いた。これについては後述する。

今回与えた課題は

- ・ 配置された「実行」ボタンをクリックすると C5 セルの背景が黄色くなるから始めた。コードとしては

```
Cells(5,3).Interior.ColorIndex = 6
```

であるが、この 1 文でセルの指定の仕方が (y 座標, x 座標) の形式であることや、数字の 6 が黄色を表すことなどが読み取れば十分である。このあと、生徒は座標を変えて指定したセルが変化することを確認したり、色コードを変更したりするなどし、ある数字以上の色コードではエラーが出るなどを見つけた生徒もいた。エラーが出た際にどのように復旧するかなどの方法も体験的に獲得し、仲間に広めていくなどした。ただし、実行の前に、着色したセルの背景色を白に戻すのはコマンドメニューからの手動操作による生徒が多い。

続いて、「実行」ボタンに

```
Cells(3,3).Interior.ColorIndex = 6
```

```
Cells(4,3).Interior.ColorIndex = 6
```

を割り当て、一つのボタンで二つの命令が実行されることを体験させた。当然のように、

```
Cells(5,3).Interior.ColorIndex = 6
```

```
Cells(6,3).Interior.ColorIndex = 6
```

```
Cells(7,3).Interior.ColorIndex = 6
```

```
:
```

```
:
```

といコードを入力する生徒があらわれ、広まっていった。そこで、マルチステートメントを指導し

```
Y = 3 : Cells(Y,3).Interior.ColorIndex = 6
```

という表現を紹介し

```
Y = 3 : Cells(Y,3).Interior.ColorIndex = 6
```

```
Y = 4 : Cells(Y,3).Interior.ColorIndex = 6
```

```
Y = 5 : Cells(Y,3).Interior.ColorIndex = 6
```

```
:
```

```
:
```

```
Y = 11 : Cells(Y,3).Interior.ColorIndex = 6
```

で、パズル面の C 列がすべて黄色くなることを体験させた。これをもとに、繰り返し処理の

```
For Y = 3 to 11 Step 1
    Cells(Y,3).Interior.ColorIndex = 6
Next Y
```

を紹介することで、スムーズに繰り返し処理の構文が定着した。生徒は、X 座標についてもコーディングし、横方向の繰り返しに成功するところまで全員到達した。縦横に着色したいという生徒からの要望があったが、ループの入れ子構造については次回の内容とした。

### 第 07 回 (2007/10/25) ...ループの入れ子構造

まず、前回の復習として、「B2 セルから J2 セルまでをグレーにする」題材を与えた。続いて前回回避したループの入れ子構造について、右のような構造を示し、これをもとに、

```
For X = 3 to 11 Step 1
    For Y = 3 to 11 Step 1
        Cells(Y, X).Interior.ColorIndex = 6
    Next Y
Next X
```

```
X = 3
  For Y = 3 to 11 Step 1
    Cells(Y,X).Interior.ColorIndex = 6
  Next Y
X = 4
  For Y = 3 to 11 Step 1
    Cells(Y,X).Interior.ColorIndex = 6
  Next Y
X = 5
  For Y = 3 to 11 Step 1
    Cells(Y,X).Interior.ColorIndex = 6
  Next Y
  :
  :
  :
```

を理解させた。この結果は、「リセット(パズル面全体を白に戻す)」機能を持つボタンの必要性を生徒が発見することにつながり、新しい課題として全員が共有し全員が実装できた。Step 1 は省略可能であるが、後の指導内容を考慮し、敢えて記述した。またコーディングにおけるインデントの意味についても説明した。

### 第 08 回 (2007/11/01) ...分岐処理

分岐処理について指導した。題材は「B2 セルから J10 セルまでをグレーにする。ただし左上から右下への対角線上のセル(B2、C3、D4、...、J10)は黄色にする」とした。指定された対角線上のセルは、他のセルと異なるどのような性質を持つのかを考えさせ、x 座標と y 座標が等しいということに気付かせた。これをもとに条件の記述の仕方を指導した。

```
For X = 3 to 11 Step 1
    For Y = 3 to 11 Step 1
        If X = Y then
            Cells(Y, X).Interior.ColorIndex = 6
        Else
            Cells(y, x).Interior.ColorIndex = 15
        End IF
    Next Y
Next X
```

続いて、逆対角にあるセルを着色するプログラムを作らせたが、「x 座標と y 座標の和が 14 になるセル」という性質はすぐに見抜ける生徒が多く、全員が正しい結果を得ることができた。さらに、初期化した後で「すでになにか数値の入っているセルをグレーにする」題材を与えた。「空白ではない」「0 より大きい」など、さまざまな表現が考えられるが、何度かの失敗の後、全員が結果を得ることができた。Cells(Y, X) = " "、という Null を使った条件の書き方や、Cells(Y, X) <> " " といった比較演算子の説明をした。

## 第 09 回 (2007/11/08) ...引数

前回まで使っていたボタンは「初期化」ボタンとし、今回新しいボタンを盤面に配置させ、そのボタンに「パズル面上で数字の 1 を見つけて黄色にする」機能を持たせる題材を与えた。これは前回までの復習として、若干の試行錯誤で全員がこなせた。つづいて、「パズル面上で数字の 2 を見つけて黄色にする」「パズル面上で数字の 3 を見つけて黄色にする」といった機能を考えさせると、全員が盤面に 10 個のボタン(「リセット」「1」「2」「3」...「9」)を配置した。当然、それぞれのボタンの機能のコーディングが必要になるが、1 文字違うだけのソースを作る無駄を考えさせ、変数の概念を指導した。すなわち、盤面のパズルに影響しないセル、例えば N3 セルに対象の数字を入力し、「指定したセルに入力された数字を見つけて黄色にする」機能を考えさせた。

```
If Cells(Y, X) = Cells(3,14) then
    Cells(Y, X).Interior.ColorIndex = 6
End If
```

## 第 10 回 (2007/11/15) ...ループ内でループ変数の値を変化させてはならない

まず、前回完成させた「指定したセルに入力された数字を見つけて黄色にする」機能に、前処理として、「一旦すべてのセルを白くする」「何か数字が入っているセルをグレーにする」機能を持たせた。続いて、「指定したセルに入力された数字を見つけ、その数字が含まれる行全体をグレーにする」課題を与えた。処理としては「数値を見つけた行を固定し、列番号を 2 から 10 まで変化させながらセルをグレーにする」ことになる。新しいループを作ることはほとんどの生徒がすぐに思いつけた。しかし、行の値は y 座標の値をそのまま使えるが、列の値について改めて別の変数を用意し、その変数でループさせるという発想がなかなかできない。この課題全体の最初の難所であった。デバッグツールの利用も考えられるが、いたずらに混乱させるだけなので、変数の遷移を詳しく図解することで乗り切った。これが理解できた生徒は、「指定したセルに入力された数字を見つけ、その数字が含まれる列全体をグレーにする」課題もスムーズにできた。

```
If Cells(Y, X) = Cells(3,14) then
    For I = 3 to 11
        Cells(Y, I).Interior.ColorIndex = 6
    Next I
Else
    :
```

この時点で、パズル面には

- ・ 初期化 (Initialize)
- ・ ターゲットの数字を見つけて黄色にする (Target\_Search)
- ・ ターゲットの数字がある行をグレーにする (Row\_Check)
- ・ ターゲットの数字がある列をグレーにする (Column\_Check)

の 4 つのボタンが配置された。( ) はマクロ名である。

## 第 11 回 (2007/11/22) ...分割統治法の考え方とブロック処理

ここで、これまでに身につけた技術を整理し、今後何が必要かを考えさせた。

まず、このパズルを解く過程で、

- ・ ターゲットとした数字を入れることができないセルを見つける
- という課題の設定がある。この課題を細分化すると、



1. すでに何か数字が入っているすべてのセル
2. ターゲットとした数字を見つけたセルが属する行全体
3. ターゲットとした数字を見つけたセルが属する列全体
4. ターゲットとした数字を見つけたセルが属するブロック全体

が、課題を翻訳したものと言えることを説明し、このように、一見解決不可能な課題を、細分化された解決可能な課題の集合体としてとらえ、それを再構築してもととの課題の解決に向かう方法を、分割統治法と称することに触れた。

細分化した課題のうち、2と3は前回に解決している。また1はそれまでの課題の組み合わせで容易に解決できるので、「ターゲットの数字を見つけて黄色にする」ボタンを「すでに何か数字が入っている」ボタンにキャプションを変え、マクロ名も Not\_Blank として作り直させたが、全員がすぐに結果を得た。

今回は4に取り組むことに重点を置いた。ただし、ここではいきなりコーディングすることを避け、どのような発想でブロックの処理をするのかを討議し、図で表現させてみた。ただし、フローチャートの記号などは指導おらず、生徒が思い思いに書いた処理を丸で囲んだものから矢印が出ている程度のものである。ブレーンストーミングのような状態になったが、最終的にまず解決すべき課題が

- ・ ターゲットとした数字を見つけたセルが属するブロックの左上隅のセルの座標を見つけること

となった。これらの座標さえ見つければ、そこから右へ3つ、下へ3つの全部で9つのセルをグレーにすることは容易であるという結論に達した。これまでに指導した内容だけで生徒は、

```

If Cells(Y, X) = Cells(3,14) then
  If X < 6 then
    Anchor_X = 3
  Else IF X >8 then
    Anchor_X = 9
  Else
    Anchor_X = 6
  End If
End If

```

というコードを作ることができた。Anchor\_X は左上隅の横座標を表している。一人だけ、この内側の条件分岐の7行が

$$X = \text{Int}(X / 3) * 3$$

で表すことができる概念を捉えた生徒がいたが、Int を知らないのでは式にすることはできなかった。パズル面を C3 を起点にしたのは、このような生徒が現れるかどうかを見つけたかったからであるが、まったく不可能なことではないと考えられる。この考え方は、

X	3	4	5	6	7	8	9	10	11
X/3	1	1.33	1.66	2	2.33	2.66	3	3.33	3.66
Int(X/3)	1	1	1	2	2	2	3	3	3
Int(X/3)*3	3	3	3	6	6	6	9	9	9

という表にすることで、他の生徒の理解も得られた。

仮に A1 にパズル面の起点をおいても、

$$X = \text{Int}((X+2)/3) * 3 - 2$$

で解決できるが、これはハードルが高すぎるように感じた。しかし生徒の数字に対する感覚のレベルによっては挑戦すべき内容としてとらえてよいと考えられる。

## 第 12 回 (2007/11/29) ... 機能の統合 (サブルーチンとしての呼び出し)

前回までに完成させた 4 つのマクロ

- ・ 初期化 (Initialize)
- ・ ターゲットの数字を見つけて黄色にする (Target\_Search)
- ・ ターゲットの数字がある行をグレーにする (Row\_Check)
- ・ ターゲットの数字がある列をグレーにする (Column\_Check)

を、一つのボタンの機能として呼び出す概念を指導した。マクロコードが、ボタンからだけでなくほかのマクロからも呼び出せるという理解でよいと思われる。また、自分自身からも呼び出せることに言及すれば、再帰的なプログラムを記述することもできると思われるが、ここでは立ち入らなかった。

課題としては、「実行」というキャプションのボタンを配置し、他のボタンに割り当てられたマクロを「実行ボタン」に順に記述していった。さらに不要になったボタン群を削除させた。

これで、一応の目標を達成したことにし、さらに加えたい機能があれば自由にカスタマイズするように指示して、コンピュータ部の生徒を対象とした予備実践を終えることにした。

### (2) 「マルチメディア表現」選択者を対象にした予備実践

コンピュータ部の生徒を対象とした実践から、次年度に向けて一般の生徒全員に対する実践につなげようと考えていたが、「『数学的な興味・関心の高いと思われる少人数のコンピュータ部員』に対する実践と『数学的興味・関心にばらつきのある多人数の一般生徒』に対する実践では、質・量において大きな違いがあり、試行結果が参考にならないのではないか」という危惧があった。この点は後日、中間報告に対して、複数の審査委員から同様の指摘をいただいた。そこで、研究代表者が担当する専門科目「マルチメディア表現」の選択者について同様の実践を試みることにした。この科目は、2年次生と3年次生の合同科目であるが、3年次生が登校しない2月に、2年次生だけを相手に4回8時間の授業で行った。なお、実践に当たっては、シラバスに記述している科目の本来の内容と異なるため、生徒自身や保護者、学校長および教務関係の了解を取り付けた。

人数は13名で、少人数であることはコンピュータ部部員に対する実践と変わらないが、数学的な興味・関心という点では、まったく異質な集団と考えてよく、ほぼ全員がパズのルールを知っていたとはいえ、解こうとした生徒は少なく、また数字対して嫌悪感を持つ生徒さえいた。

進度に大きな差が出るのが予想されたため、テキスト(巻末資料参照)を作成し、一人ひとりがほぼ自習形式で課題を消化できるようにしてみた。この点は、生徒のアイデアから新しい課題を設定していくというよさを奪ってしまったが、課題の流れは、コンピュータ部部員と相談して、無理な飛躍のない順序で並べた。実践が終わって生徒に尋ねたところ、「いかにも次はこうしたいと思う課題が次に待っていた」という感想を得たので、課題の配列に大きな間違いは少なく、多人数であっても当てはめられるのではないかという感触は得た。

13人全員が、当初の課題をすべてこなし、授業後の感想では、パズルの解法を通じたプログラミングに興味をかきたてられたというものがあり、理詰めのお考えが苦手な大半の生徒にも受け入れられたことがわかった。ただし、もともとこのパズルを楽しんでいた生徒からは、このようなコンピュータに頼った解法は邪道であり、プログラムが完成していくにつれて、パズルの面白さが減少していったという指摘があった。

これは、プログラムの指導を集中的に行うあまり、手作業でパズルを解くという過程を簡略化し、十分な時間を割かなかつたためと思われる。もともとパズルに興味を示す生徒が少なかったため、このような措置をとったが、パズルの面白さを理解したからこそ解るプログラミングの楽しさという点を重視し、より多人数に実践するには注意しなければならない点であろう。それでも、プログラミングに全く関心を持たない、あるいは嫌悪感さえ持つ生徒が、徐々に没頭していく過程は、パズルには申し訳ないが、プログラミングの教材としては、肯定的にとらえてよいのではないかと強く感じた。

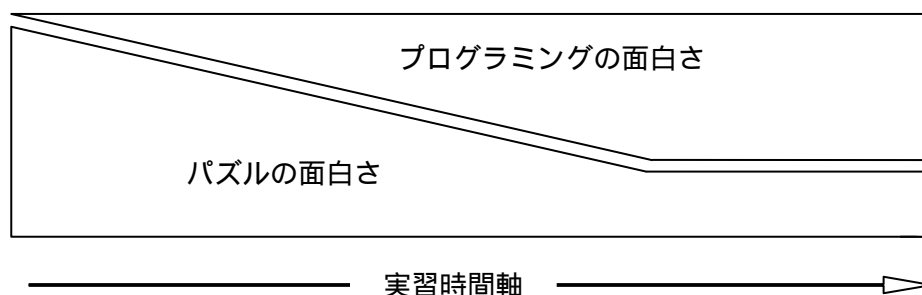


図4 面白さの遷移

#### 4 . 教材の開発

ここまでの実践でこの単元を指導するにあたって、

- ・ パズルの解法を知らない生徒への指導
- ・ パズルを解くことに面白さを感じさせる体験
- ・ 解法がパターン化されたものであることへの気づき
- ・ 直感ではなく、理詰めで物事を進める体験
- ・ 処理の自動化への期待

といったものが、プログラミングの指導の難しさを取り除く要素であると考えられた。これらを一つひとつ解決するために、次のような計画を立てた。

1. 4月の早い段階に数独の解法を指導し、自分自身で最後まで解けたという体験をさせる。
2. 授業の内容とは無関係に、毎回課題として数独の問題を与え、このパズルを身近なものにする。
3. 4月当初から、コンピュータの操作について、指示された操作を間違いなくたどれることとは別に、その他の操作を自由に試す時間を意図的に設け、少々の混乱を引き起こしても、取り立てて問題としない雰囲気を作る。
4. Excelのセルを着色する機能などを、他の単元で経験しておく。

また、この単元を指導するにあたって、他の担当者の中にはプログラミング経験のない教師もあり、適切なテキストの開発が必須であることもわかった。マルチメディア表現の選択者に対するテキストも完全なものではなく、問題点はその都度、改善した。結果として、テキストは数独の解法の解説をするものと単元を展開するためのものの2分冊とし、リング製本して実習中に広げた状態で置くことができるようにした。

また、校外における学会や研究会等での発表も積極的に行い、そこで関心を持たれた先生方に実際に授業で使っていただき、指摘される問題点についても検討し、さらにテキストに反映させた。この作業は、1年生全員への授業が終わった後も続き、今なお手を加えている。意見をいただく機会を得た団体は、巻末にまとめた。テキストは、<http://www.darumakai.com/sudoku/koduki/> からダウンロードできる。

## 5. 1年次生必修修科目「情報C」における実践

1年次生の必修修科目である「情報C」は、本校では週に1回、2時間連続で行われる授業である。この授業の4回8時間をこの単元の学習にあて、10月から11月にかけて実践した。なお、前述したとおり、数独を解き切る体験や、問題のプリントを何度も配って数独を身近なものとして気軽に解こうとすることは4月当初から授業に埋め込んだ。

まず、慣れ親しんだ数独の解法を、分割統治法で捉えなおすことから授業を始めた。生徒にとっては、こうした用語が難解に聞こえるようであるが、図5を使った説明で、日常的に行っている行為であることに気づかせることができた。

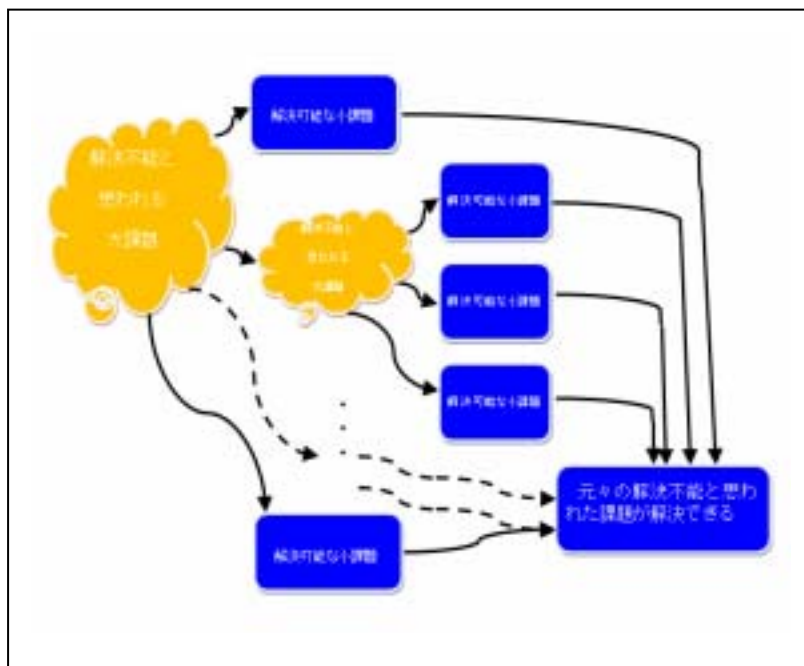


図5 分割統治法

テキストを使った実習が始まると、生徒によって進捗状況に差が出るのが予想されたため、各課題のあとに演習問題を置き、速く課題を終える生徒に自習させたが、これは逆に理解の速い生徒の理解をより深め、理解の遅い生徒が時間に追われるという結果になってしまった。しかし、理解の遅い生徒も、放課後や自宅で課題をこなすなど、途中で投げ出す生徒はいなかった。授業後の生徒の感想によると、これらは

- ・ 課題が細分化されており、短いスパンで次々と達成感が得られること、
- ・ 結果が視覚的に得られ、興味が持続したこと
- ・ 次の課題との接続性に無理がなく、途中でやめようと思わなかったこと

などが挙げられていた。単にテキストをなぞるだけでは、入力の違いが遅い進捗に影響することになるが、演習問題で理解の定着を図ったことが、功を奏したと思われる。

一部の速い生徒は2回目の途中で設定した課題を終えてしまった。これらの生徒には、あとどのような課題が残っているか、それをどのように解決するかを考えさせ、自由に作業をさせた。この単元では、ひとりひとりが自分で考えて試行錯誤することを重視したため、作業の滞っている生徒を手伝うことは一切させなかった。それでも、4回8時間の授業ですべての生徒が設定した課題をこなした。また、この単元が終わった後も、放課後にコンピュータ教室にきて、発展的な課題を見つけて取り組むグループも見られた。

## 6. 生徒の反応・変容

### (1) 生徒の感想から

授業実践の最中、あるいは実践が終わってから聞かれた生徒の感想をまとめると、

- ・ プログラミングという一見寄り付きがたい領域に踏み込んだが、おもしろい。
- ・ プログラミングは理系の人だけができればいいものだと思い込んでいたが、分割統治法の話はプログラミングとは別に、さまざまな問題解決に活かせる手法だと思った。
- ・ 全然わからない内容ではなく、頑張ったらできる内容なのでおもしろかった。

- ・ プログラムを間違えて、変な場所のセルに色がついたりしたけど、その失敗もおもしろかったし、どこにエラーがあるかを探して直す過程がおもしろかった。
- ・ 普段はメールか Web ページを見るしか使わない家のコンピュータでもやってみた。家でもやってみたらできておもしろかった。
- ・ どこにエラーがあるかすぐに見つける先生はすごい。
- ・ プログラミングに行き詰ったら鉛筆と消しゴムでパズルを解いて頭をリフレッシュしたが、結局ずっと頭を使っていたことになって疲れた。

などが得られた。ほぼ肯定的にとらえられたと考えてよいと思われる。

## (2) 教師の感想から

授業は各クラス2名の教師が受け持っている。1名は情報の免許を持つ教師であるが、もう1名は情報の免許を持たない教師で、数学科、理科、英語科、芸術科など多岐にわたる。またこのうち、体系的なプログラミング教育を受けた経験のある教師は、研究代表者だけである。その他の教師は、独学でプログラミングの経験のある者が3名、全くプログラミングの経験のない者が6名である。

プログラミングについて、単元として構成し授業に組み込んでいくことには、すべての教師に不安や戸惑いがあり、その理由は「生徒に指導する前に自分がわかっていない」ことや「総合学科における情報教育においてそもそもプログラミングの指導の必要性を感じない」といったものまでであった。また、以前にモンテカルロ法による円周率の算出を表計算ソフトで指導した経験のある教師から、うまくいかなかったという指摘もあった。これらに点について、例えば円周率の課題とは違って、「素材に生徒が興味を示さない」ということが回避できそうであること、また他の課題解決学習に分割統治の考え方が応用でき、総合学科の集大成としての卒業研究への取り組みと乖離しないことなどを挙げるとともに、出来上がったテキストや授業で使うライドなども提示した。

実際に授業をやったあとの感想では、「予想以上に生徒が積極的に取り組んだ」という感想が多く、また教材も平易で連続性があり、プログラミング学習の導入教材としては悪い評価はできないという意見をいただいた。また、配布されたファイルに設定済みのボタンの配置や罫線での修飾などから生徒にやらせるべきであるという意見も出たが、これは教材観の違いによるものと思われ、生徒の感想にあったこの授業を評しての「3分後にはプログラマー」という表現は、プログラミングの最初の段階で、ボタン配置などの寄り道をしていては出てこない感想ではないかと思われる。

## (3) 授業以外で

わずかな人数であるが、希望者が放課後にコンピュータ教室で、あるいは家庭でテキストの続きとなる課題を自ら設定し、解決に向けて取り組んだという報告を聞いた。それらのほとんどの生徒が、「決定できる数字を見つけて自動的に埋め込む」という課題に取り組んだようである。考え方の相談に来た生徒の中には、アイデアとして「関数の再帰的定義」を持った生徒も現れたが、終了判断のフラグを立てたり大域脱出などの説明から始める必要があり、完全な理解には至らなかった。

最終的にはボタン一つでパズルを自動的に解くプログラムを作った生徒もいた。また今回の授業内容を土台にして彼が作ったこのプログラムでは、解ける問題と解けない問題があり、「推論が必要な問題は解けない」というところまで見つけた生徒もいた。バックトラックアルゴリズムなどの指導にもつながる発想であると思われる。こうした発想は大事にしたいと思う。

## 7. 成果と課題

この研究テーマについては、アイデアの段階からいくつかの学会や研究会、セミナーなどで提案し、その都度、さまざまなご意見をいただいていた。初期の段階は、高等学校段階の情報教育においてプログラミング教育が必要であることと、指導者が満足なレベルではないことのギャップをどう埋めることができるかという内容であった。この研究課題を具体化し、授業実践を始めてからは、使用するテキストの内容や構成について意見をいただき、より効果的なテキストの完成に役立った。授業の実践を終えてからの発表では、生徒の受け止め方や他のレベルの教師、他のレベルの生徒での実践の可能性について議論いただいた。

途中、学会や研究会の参加者の紹介で、いくつかの都道府県の教育委員会の招きで、情報科の担当者の勉強会で講演をしたり、ワークショップを開くなどの機会も得た。

そうした場で評価されたことは、生徒にとっては、この実践で取り組んだ課題は平易であり、少人数で興味ある生徒の集まりから、興味・関心のばらつきの大きい多人数まで、誰ひとり脱落することなく実践できたことである。これはシンプルなルールのパズルであること、パズルと Excel の親和性が高いこと、結果が即座に視覚的に確認できること、パズルの解法がプログラミングの基礎的な考え(逐次処理・繰返し処理・分岐処理)の複雑でない組み合わせだけで構成されることなどに起因すると思われる。現状では、この題材がプログラミング指導に適していることは間違いないと実感している。

手元にある自分自身が受けた 1980 年代後半のプログラミング教育の導入部分を見返してみても、

- ・ 1 から 10 までの自然数の和を求めなさい
- ・ 底辺と高さを入力して、三角の面積を求めなさい。
- ・ 摂氏と華氏の変換をなさい。

などが見つかる。おそらく最後の例などはアメリカの教材の翻訳であろうが、日本で普通の生活をしているなかで、華氏を使うことはほとんどない。研究会でも「華氏はこの課題のときに始めて存在を知ったが、その課題以降、使ったことがない」という声も出た。また全体を見ても、プログラミングの文法体系にのっとったものではあるが、課題そのものが系統的なものではなく、結果を得ても「それで？」というものが多い。この点についても、現在のプログラミング学習の最初の体験として約束事のように「Hello World!」を表示させることに疑問を持っておられる方が多いことがわかった。こうした点を見ても、この研究のテーマは、広く受け入れられるものであると確信している。ただし、まず、これが正しい指導方法であるかどうか、プログラミング指導のツボをすべて押さえているかどうかは別である。

例えば、この実践でのプログラムのコーディングにあたって、理解の度合いがインデントの置き方に表れていると感じており、今後の一つの視点として注目したい。例えば、理解度の高い生徒は、コーディングに際して、最初からインデントが置ける。すなわち、for ~ next や if ~ else ~ end if の間に挟まれる処理を、正しくインデントを置いて入力し始める。それに対して、理解度の低い生徒は、入力した後でつじつまを合わせるようにインデントを置く傾向がある。あるいは、インデントを置いたソースを提示して、そのまま入力するように指示しても、インデントを置けない、すなわち「行頭の空白が見えない」という生徒が存在する。VBE では直前の行と行頭をそろえる設定がデフォルトであるので、この機能を解除してインデントの置き方の理解とプログラミングの理解の関係についても考えてみたい。

研究会などで発表すると、よく「推論が必要な問題に対応できるレベルまで行き着くか」「数独の問題を作るプログラムに発展するか」などの質問を受けることがある。多くは数独に魅せられた方からの発言であるが、本研究が「プログラミングの導入部分で生徒の興味・関心を引きつける教材を作り、『情報の科学的な理解』をより必要なものとして感じさせること」にあり、決して「どんなレベルの数独の問題でも解けるプログラ

ムを作ること」ではないことを理解していただきたいと考えている。生徒に提示した数独の問題も、バックトラックなどの思考過程を必要とするレベルの問題は排除し、また複数の解を持つ問題も排除した。これらはより高いレベルのプログラミングの要素をもち、現段階で、すべての高校生全員に理解を求めるものではないという判断をしたからである。

今回の研究では、当初は「決定できる数字を見つけて自動的に埋め込む」ところまでを単元として、授業で取り組む予定にしていた。しかし、他の単元との関係において、十分な時間が取れないことから、「分割統治法」が体験でき、「構造化プログラミング」を体験する最小限の要素に絞って実践することになった。したがって、この「決定の自動化」の部分は、興味ある生徒の自由課題になってしまったことは残念である。将来的に、「科学的な理解」の比重が高まり、十分な時間が割けるときに向けて、テキストの開発を続けていきたい。さらに、よりプログラミングの入門書としての完成度を高めるためには、文字列操作を習得させる部分が必要になると思われるが、これは、冒頭に紹介した解法2を利用することで解決可能であることがわかっている。これもまた今後のテキストの開発に組み込んでいきたい。

#### 【参考文献】

- 文部省：高等学校学習指導要領解説 情報編、開隆堂、2000  
大村あつし：かんたんプログラミング Excel VBA 基礎編、技術評論社、2004  
大村あつし：かんたんプログラミング Excel VBA コントロール・関数編、技術評論社、2004  
大村あつし：かんたんプログラミング Excel VBA 応用編、技術評論社、2005  
棚床弘樹：鉛筆パズルゲームプログラミング、ソフトバンククリエイティブ株式会社、2007  
ニコリ：ポケット数独 初級編、株式会社ニコリ、2006  
早乙女和宏ほか：30 時間でマスターExcel VBA、実教出版株式会社、2003

#### 【発表する機会を得た学会・研究会等】

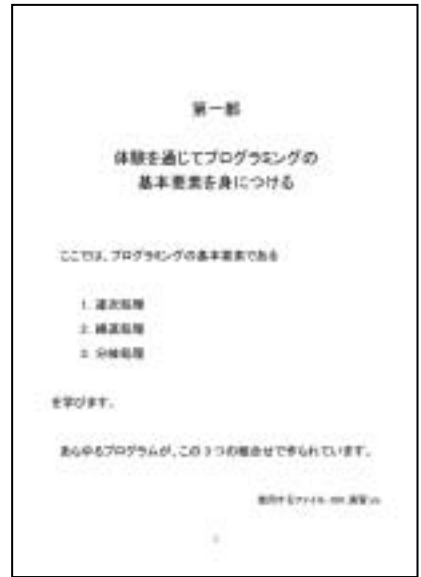
- ・ 日本情報科教育学会第 1 回全国大会(滋賀大学)
- ・ 実教出版 授業展開セミナー(札幌・東京・福岡)
- ・ 第 39 回 ICTE 情報教育セミナー(関西大学)
- ・ 情報コミュニケーション学会第 6 回全国大会(園田学園女子大学)
- ・ SITE2009(Society for Information Technology and Teacher Education) (優秀ポスター賞受賞)

#### 【研究協力者】

テキストを使って授業をしていただいたり、参考となる意見をいただいたりしました。

- ・ 石川県立金沢泉丘高等学校(鹿野利春先生および生徒の皆さん)
- ・ 兵庫県立西宮今津高等学校(佐藤万寿美先生)
- ・ 千里金蘭大学(高橋参吉教授)









## 第二部

### パズルの解法を視覚化する

ここでは、第一冊で学習したプログラミングの基本事項の

1. 変数宣言
2. 数値計算
3. 分岐処理

を組み合わせ、パズル「数独」の解法を視覚化します。

読者のみなさんへ

### 方針の整理

- 「1」が空いているマス
- 「2」が空いているマス
- 「3」が空いているマス
- 「4」が空いているマス

- 「1」が空いているマスを探る
- 「2」が空いているマスを探る
- 「3」が空いているマスを探る
- 「4」が空いているマスを探る
- 「1」が空いているマスを探る
- 「2」が空いているマスを探る
- 「3」が空いているマスを探る
- 「4」が空いているマスを探る

### 読解 7

#### 「1」が空いているマスを探る

1. 読解
  - 「1」が空いているマスを探る
2. 読解
  - 「2」が空いているマスを探る

### 読解 8

#### 「2」が空いているマスを探る

1. 読解
  - 「2」が空いているマスを探る
2. 読解
  - 「3」が空いているマスを探る

### 読解 9

#### 「3」が空いているマスを探る

1. 読解
  - 「3」が空いているマスを探る
2. 読解
  - 「4」が空いているマスを探る

### 読解 10

#### 「4」が空いているマスを探る

1. 読解
  - 「4」が空いているマスを探る
2. 読解
  - 「1」が空いているマスを探る

### 読解 11

#### 「1」が空いているマスを探る

1. 読解
  - 「1」が空いているマスを探る
2. 読解
  - 「2」が空いているマスを探る

### 読解 12

#### 「2」が空いているマスを探る

1. 読解
  - 「2」が空いているマスを探る
2. 読解
  - 「3」が空いているマスを探る

### 読解 13

#### 「3」が空いているマスを探る

1. 読解
  - 「3」が空いているマスを探る
2. 読解
  - 「4」が空いているマスを探る

**課題 12**

「1」以外の数字をすべて1に置き換える

1. 解説

課題11の課題はそのままのままで、1以外の数字をすべて1に置き換えること。1以外の数字はすべて1に置き換える。

3. 実行

- 1) 実行結果を確認する
- 2) 「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する
- 3) プログラムを完成させる

```

#include <stdio.h>
int main(void)
{
    int i, j;
    for (i = 0; i < 10; i++)
        for (j = 0; j < 10; j++)
            printf("%d ", (i * 10 + j) % 9 + 1);
    printf("\n");
}

```

4. 課題2

- 1) プログラムの作成
- 2) 「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する

**第三部**

**プログラムを改善する**

ここでは、第2章で作成したプログラムの改善点として

- 「1」以外の数字についても処理可能にすること
- およびそれにより発生する問題の解決策

を解説します

遠藤裕之(アイン)の講座 14

**課題 13**

「1」以外の数字をすべて1に置き換える

1. 解説

「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する。

2. 実行

- 1) プログラムの作成
- 2) 「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する

4. 課題2

- 1) プログラムの作成
- 2) 「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する

**課題 14**

数字の重複をなくす

1. 解説

課題13の課題はそのままのままで、数字の重複をなくすこと。数字の重複をなくすこと。

2. 実行

- 1) プログラムの作成
- 2) 「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する

```

#include <stdio.h>
int main(void)
{
    int i, j;
    for (i = 0; i < 10; i++)
        for (j = 0; j < 10; j++)
            printf("%d ", (i * 10 + j) % 9 + 1);
    printf("\n");
}

```

4. 課題2

- 1) プログラムの作成
- 2) 「1」以外の数字をすべて1に置き換えるプログラムを作成して実行を確認する

では、このツールを使って数独を解いてみましょう

**今後の課題予定**

- 課題15の課題はそのままのままで、プログラムの改善点として
- 「1」以外の数字についても処理可能にすること
- およびそれにより発生する問題の解決策

**評価**

- 課題15の課題はそのままのままで、プログラムの改善点として
- 「1」以外の数字についても処理可能にすること
- およびそれにより発生する問題の解決策

テキストおよびその他の資料は <http://www.darumakai.com/sudoku/koduki/> からダウンロード可能です。