

# 情報関係基礎のプログラミングの問題を解いてみよう

名古屋高等学校 中西渉

2018年12月27日

## 1 実行環境について

### 1.1 PEN

大学入試センター試験の情報関係基礎という科目では、プログラミングの問題に DNCL という言語を用いています。これはもともと日本語をベースにした擬似言語だったのですが、これを実行できる環境が大阪学院大学情報学部の西田研究室と大阪市立大学大学院創造都市研究科の松浦研究室の共同プロジェクトである初学者向けプログラミング学習環境 PEN (\*1) という形で開発されました。ここで使われている言語は DNCL を拡張した xDNCL です。何しろ DNCL には入力さえないので...

### 1.2 WaPEN

その後、フローチャートから xDNCL のコードを生成する機能を付け加えた PenFlowchart と、それと同じことが Web ブラウザ上でできる環境 WaPEN を中西が開発しました (\*2)。今回の実習では WaPEN を使います。

Web ブラウザを立ち上げて <https://watayan.net/misc/ipsj2018/> にアクセスしてください (\*3)。左の画面でコードを入力して、実行ボタンを押すとプログラムが実行されます。右の画面でフローチャートを作ると（縦棒の上で右クリックするとメニューが表示される）即座にコードに反映されます。逆にコードをフローチャートに反映させるには「コード → フローチャート」ボタンを押す必要があります。いくつかのサンプルプログラムがあるので、呼び出して感触をつかんでみてください。

```
1 整数 a,b
2 a←0
3 b←random(8)+1
4 「1から9の数字を当ててください」を表示する
5 繰り返し,
6 | aを入力する
7 | aを表示する
8 | もしa>bならば
9 | | 「大きい」を表示する
10 | | を実行し、そうでなければ
11 | | もしa<bならば
12 | | | 「小さい」を表示する
13 | | | を実行する
14 | | を実行する
15 | を、a=bになるまで実行する
16 | 「あたり」を表示する
17
18
19
20
21
```

The flowchart on the right starts with 'はじめ' (Start), followed by 'a←0', 'b←random(8)+1', and a loop '「1から9の数字を当ててください」'. After 'aを入力' (input a), it checks 'a > b'. If 'Y' (Yes), it outputs '大きい'. If 'N' (No), it checks 'a < b'. If 'Y', it outputs '小さい'. If 'N', it outputs 'あたり'. The loop continues until 'a = b になるまで' (until a = b).

(\*1) <https://www.media.osaka-cu.ac.jp/PEN/>

(\*2) <https://watayan.net/prog/>

(\*3) このアドレスは講習が終わってからもしばらくこのままにしておきますが、適当な時期に前触れなく削除する予定です。

## 2 xDNCL の文法

### 2.1 変数の宣言

変数はプログラムの先頭で宣言しないと使えない。変数が複数あるときはコンマで区切って記述する。配列は [ ] の中に最後の番号を書いて宣言する（配列の番号は 0 から始まる）。

整数 《変数名》  
 実数 《変数名》  
 文字列 《変数名》

### 2.2 計算

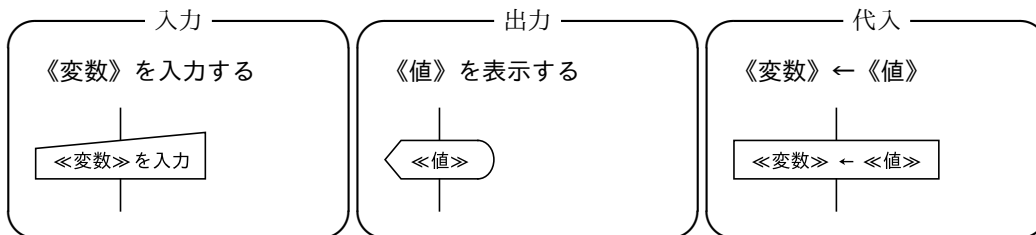
+, -, \*, /, % が使える。\* は掛け算, / は割り算を表し, % は割り算をしたときの余りを表す。なお, 整数どうしを / で割ったときは, 余りを切り捨てて計算が行なわれる。

### 2.3 条件

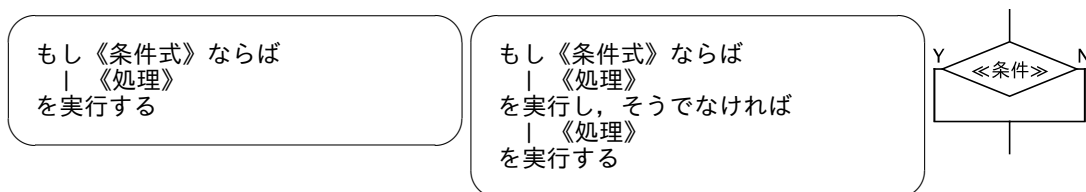
=, !=, >, <, >=, <= によって二つの値を比較する (!= は ≠, >= は ≥, <= は ≤ をそれぞれ表す)。条件と条件を ‘かつ’ や ‘または’ で組み合わせることもできる。

### 2.4 入力・出力・代入

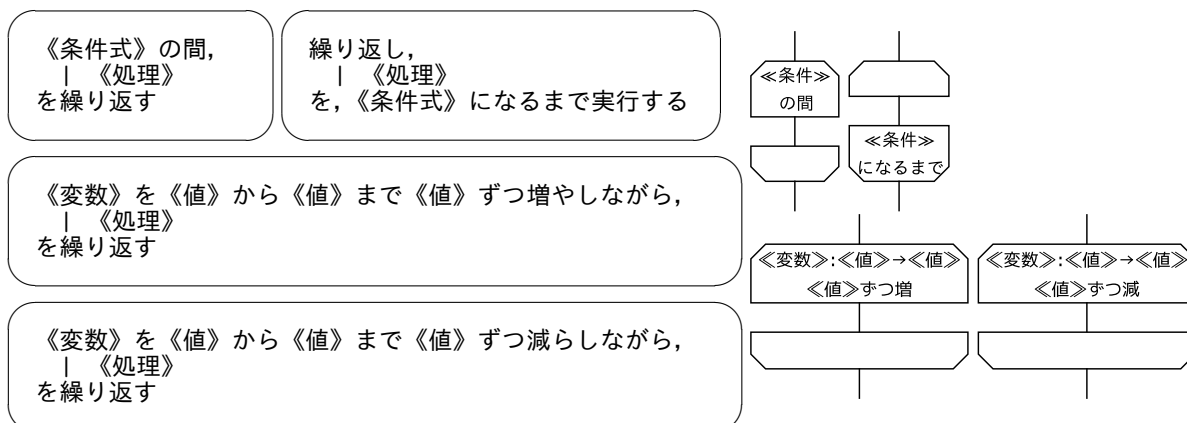
文字列は「」で囲む。複数の値や文字列を ‘と’ でつなぐことができる（例：「x は」と x を表示する）。



### 2.5 選択



### 2.6 繰り返し



### 3 手始めに

センター試験の問題に取り組む前に、ガチャのシミュレーションを作ってみましょう。賞品は 10 種類ですべて等確率で出るものとし、全種類が揃うまでに何回ガチャを引くことになるかを実験してみます。random(n) は 0 以上 n 以下の整数の乱数を発生します。

```
(01) 整数 kaisu,ransu,syohin[9],nuke,n
(02) kaisu ← 0
(03) syohin ← [0,0,0,0,0,0,0,0,0]
(04) 繰り返し,
(05) | kaisu ← kaisu+1
(06) | ransu ← random(9)
(07) | syohin[ransu] ← 1
(08) | nuke ← 0
(09) | n を 0 から 9 まで 1 ずつ増やしなが、
(10) | | syohin[n] を改行なしで表示する
(11) | | もし syohin[n]=0 ならば
(12) | | | nuke ← 1
(13) | | | を実行する
(14) | | を繰り返す
(15) | | 「 」と kaisu を表示する
(16) | を, nuke=0 になるまで実行する
```

まず (01) は変数の宣言です。すべて整数にするので「整数」ボタンを押して、変数を書き並べます。フローチャートでは、上の「整数」の空欄に、変数を書き並べます。

(02), (03) は代入です。「代入」ボタンを押して、変数名と値を入力します。フローチャートでは、縦棒の上で右クリックして出てくるメニューから「代入」を選び、表示された代入の四角を右クリックして出てくるメニューから「編集」を選んで書き込みます（以下も同様なので、フローチャートに関する説明は省略）。なお、(03) では配列にまとめて値を入力していますが、これは PenFlowchart や WaPEN による拡張です (\*4)。

次は (04)～(16) の繰り返しです。これは条件が後にきているので「～になるまで実行する」ボタンを使います。あとは繰り返しの中に (05)～(15) のプログラムを書いていきます。行先頭の「|」は字下げを表しています。

(05)～(08) は代入です。(09)～(14) は繰り返しですが、n を増やしなが、繰り返すので「増やしなが、」ボタンを使います。

(10) は表示するので出力ですが、改行なしなので「改行無出力」ボタンを使います。(11)～(13) は「もし」の構文ですから「もし」ボタンです。(12) は「代入」、(15) は「出力」です。(15) は (14) の行末で Enter を押すことで新しい行ができるので、そこに書き込みます。

完成したら実行して結果を確認しましょう。うまくいかない場合は、間違いがないか確認してください。ちなみに n 種類の賞品のガチャ（等確率）をコンプリートするための回数の期待値は  $n\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}\right)$  であることが知られています。n = 10 のときの期待値は 29.28... です。

## 4 情報関係基礎のプログラムに挑戦

### 4.1 第 3 問のプログラム

情報関係基礎の第 3 問は別紙のようにプログラミングの問題になっています。別紙 31 ページにプログラムがありますが、これをその通りに打ち込んでも結果は出ません。実はこれ以前に変数の初期化などの作業が必要です。この問題では tate, yoko, Masu に適当な値を代入しておかなくてははいけません。そこでその部分を既に入力したプログラムを、サンプルの下の「センター試験問題の前処理」に用意しました。これを呼び出して、その続きを作っていくことにします。2 次元配列変数 Masu と別紙 30 ページ図 4 は縦横が逆になっていることに注意してください。

また、WaPEN で実行できるようにするためにいくつかの修正をしないではいけません。そこで完成版のプログラムを以下に掲載します。カッコ内の行番号は別紙 31 ページの行番号に対応しています（冒頭の (00) は初期化のために付け加えた部分で、「センター試験問題の前処理」ボタンで入力できる）。変更点は次の通りです。

(\*4) 情報関係基礎の問題では初期値を配列で指定することが多いので、まとめて代入できないととても面倒です。

- (07) で2つの代入文が書かれているが、WaPEN ではこれができないので2行に分けた。
- (15), (17) で文字列のくくりを""でなく「」にした<sup>(\*5)</sup>。
- (20) の改行を「」にした。

```

(00) 整数 tate,yoko,Masu[11,9],x,y,nutta
(00) tate ← 9
(00) yoko ← 11
(00) Masu[1] ← [0,1,1,1,1,1,1,1,1,1,1]
(00) Masu[2] ← [0,1,0,0,0,1,0,0,0,0,1]
(00) Masu[3] ← [0,1,1,1,0,0,0,1,0,1]
(00) Masu[4] ← [0,9,0,0,0,1,0,1,0,1]
(00) Masu[5] ← [0,1,1,1,0,1,0,1,0,1]
(00) Masu[6] ← [0,1,0,1,1,1,0,1,1,1]
(00) Masu[7] ← [0,1,0,0,0,1,0,1,0,1]
(00) Masu[8] ← [0,1,1,1,0,0,0,1,0,9]
(00) Masu[9] ← [0,1,0,0,0,1,1,1,0,1]
(00) Masu[10] ← [0,1,0,1,0,0,0,0,0,1]
(00) Masu[11] ← [0,1,1,1,1,1,1,1,1,1]
(01) 繰り返し,
(02) | nutta ← 0
(03) | | y を 2 から tate-1 まで 1 ずつ増やしなが,
(04) | | | x を 2 から yoko-1 まで 1 ずつ増やしなが,
(05) | | | | s ← Masu[x-1,y]+Masu[x+1,y]+Masu[x,y-1]+Masu[x,y+1]
(06) | | | | もし Masu[x,y]=0 かつ s=3 ならば
(07) | | | | | Masu[x,y] ← 1
(07) | | | | | nutta ← 1
(08) | | | | を実行する
(09) | | | を繰り返す
(10) | | を繰り返す
(11) | を, nutta=0 になるまで実行する
(12) | y を 1 から tate まで 1 ずつ増やしなが,
(13) | | x を 1 から yoko まで 1 ずつ増やしなが,
(14) | | | もし Masu[x,y]=1 ならば
(15) | | | | 「■」 を改行なしで表示する
(16) | | | | を実行し, そうでなければ
(17) | | | | 「□」 を改行なしで表示する
(18) | | | | を実行する
(19) | | | を繰り返す
(20) | | 「」 を表示する
(21) | | を繰り返す

```

## 4.2 改良版のプログラム

ここまでのことを参考に、別紙 32, 33 ページにある改良版のプログラムを作ってみてください。変数に `nutta` はなくなるものの、新たに `i`, `j`, `di`, `dj` が追加されるので、冒頭の変数宣言を書き直すのを忘れないようにしてください。

### おわりに

同じ働きをするプログラムはいろいろな言語で表現することができます。高校の情報の教科書では JavaScript や Excel VBA が主に使われています。特にセルを配列と見なして処理すると、表示部分を作らなくてもいいので楽です。実際、簡単な数独を VBA で解くプログラムが実教出版の教科書で扱われていましたが、そのような方法を用いていました。

多くの言語の中から私が勤務校で xDNCL を 10 年以上使い続けているのは、キーワードが英語であることへの抵抗感を感じずに済むことや、他の多くの手続き型言語への翻訳がしやすいこと、筆記試験がやりやすいことなどがその理由です。これを用いている教科書は現状ありませんが、「翻訳」をする手間を厭わなければ十分使えるものだと考えています。

(\*5) 実は""でも正しく動作する。