

1 E 情報処理 講義ノート

【第2版】

作成者

桐島 俊之

独立行政法人 国立高等専門学校機構

奈良工業高等専門学校

電気工学科

Department of Electrical Engineering

Nara National College of Technology

Institute of National Colleges of Technology, Japan

目 次

1. アンケート	1
2. マニュアル計算の練習	3
3. 算術代入計算の練習	6
4. 統計計算の練習	9
5. プログラミングの基礎	11
5.1 BASIC言語によるプログラミングの基礎	11
5.1.1 BASICとは何か?	11
5.1.2 プログラムとは何か?	11
5.1.3 プログラミングの基本はどう習得するのか?	11
5.2 BASICプログラムを実行してみよう	12
5.2.1 画面に文字列を表示するプログラム	12
5.2.2 2変数の足し算を行うプログラム	12
5.2.3 足し算練習プログラム	12
5.2.4 円の面積を計算するプログラム	13
5.2.5 お楽しみプログラム (その1)	13
5.2.6 お楽しみプログラム (その2)	13
5.2.7 お楽しみプログラム (その3)	14
6. BASIC言語でプログラミング	15
6.1 BASICで統計計算をするプログラムを作ろう	15
6.2 BASICでキャラクタを表示しよう	16
6.3 BASICで数当てゲームを作ろう	17
6.4 キャラクタを動かそう (その1)	18
6.5 キャラクターを動かそう (その2)	19
6.6 追いかけて (られる) ゲームを作ってみよう	20
6.7 追いかけてゲームのプログラム	22
6.8 インベーダーゲームもどきを作ろう	25

6.8.1	初期化	25
6.8.2	各種情報の表示	25
6.8.3	メインルーチン	26
6.8.4	玉の処理 (サブルーチン)	27
6.8.5	インベータの処理 (サブルーチン)	27
6.8.6	インベータの爆発処理 (サブルーチン)	27
6.8.7	ゲーム終了処理 (サブルーチン)	28
6.8.8	検討課題	28
6.9	ダンジョン探検ゲームを作ろう	29
6.9.1	ゲームのルール	29
6.9.2	ゲームの仕方	29
6.9.3	初期設定	29
6.9.4	メインプログラム	30
6.9.5	マップの表示	31
6.9.6	アイテム取得処理	32
6.9.7	拒絶処理	32
6.9.8	ゴール処理	33
6.9.9	ゲームオーバー処理	33
6.10	ブロック崩しゲームを作ろう	35
6.10.1	初期化	35
6.10.2	移動量の設定	35
6.10.3	ゲーム画面描画	35
6.10.4	メインプログラム	36
6.10.5	ボールの移動	36
6.10.6	方向転換	37
7.	C言語でプログラミング (入門編)	39
7.1	C言語によるプログラミングの基礎	39
7.2	足し算プログラムの比較 (B A S I C言語 VS C言語)	39
7.3	C言語プログラミングの利点とは?	41
7.4	参考図書	41
7.5	B A S I CプログラミングとC言語プログラミングの違い	42

7.5.1	BASIC言語によるプログラミングでは	42
7.5.2	C言語によるプログラミングでは	42
7.5.3	C言語プログラムの入力方法	43
7.5.4	C言語プログラムを実行してみよう	43
付録		I
A. 【解答】 マニュアル計算の練習		I
B. 【解答】 算術代入計算の練習		IV
C. 【解答】 統計計算の練習		VII
D. 夏休みの課題		IX
D.1	はじめに	IX
D.2	プログラムリストその1	IX
D.3	プログラムリストその2	XI
D.4	プログラムリストその3	XII
D.5	プログラムリストその4	XIV
D.6	プログラムリストその5	XVI
E. 冬休みの課題		XX
E.1	はじめに	XX

1. アンケート

本年度の情報処理の授業を進める上での参考資料としますので，以下の質問項目について自由に回答してください。

1. コンピュータに興味・関心はありますか？
2. コンピュータを利用して何をしてみたいと思いますか？
3. 電気と情報には密接な関係があると思いますか？
4. 家に利用可能なコンピュータはありますか？
5. インターネットと接続していますか？
6. 週何時間程度コンピュータを利用していますか？
7. コンピュータで何をしていますか？ 好きなだけ◎印を記入してください。
 - ホームページ関係
 - メール関係
 - ゲーム・娯楽
 - A V機器として利用
 - 表計算／ワープロなど
 - プログラミング
 - その他
8. 以下の内，知っているプログラミング言語はどれですか？ 好きなだけ◎印を記入してください。
 - FORTRAN
 - COBOL
 - PL/I
 - アセンブリ言語
 - BASIC (Visual Basic を含む)
 - C(C++) (Visual C++を含む)

- Perl
 - Java
 - その他
9. 使用している，または，使用したことのある言語はどれですか？ 好きなだけ◎印を記入してください。
- FORTRAN
 - COBOL
 - PL/I
 - アセンブリ言語
 - BASIC（Visual Basic を含む）
 - C(C++)（Visual C++を含む）
 - Perl
 - Java
 - その他
10. どのような目的で，あるいは，いつどこでプログラミングを経験しましたか？
11. プログラミングは面白そうだと思いますか？
12. プログラミングは難しそうだと思いますか？
13. ポケットコンピュータを使用したことがありますか？
14. ポケットコンピュータで何ができそうだと思いますか？
15. 情報処理の授業により高度な内容を期待しますか？ それとも入門的な内容を希望しますか？
16. 情報処理の授業で取り上げて欲しいトピックス，または，この授業に対する要望がありましたら好きなだけ列挙してください。

次回の授業までにポケットコンピュータの取扱説明書を良く読んで，授業で利用できる状態にしておいてください。なお，ポケットコンピュータは盗難発生の原因になりますので，保管や管理は各自できちんとしてください。また，動作不良が見つかったり，故障してしまった場合は，奈良高専生協まで相談に行ってください。

授業で分からなかったことなどは，オフィスアワーなどを利用して気軽に質問してください。

2. マニュアル計算の練習

- 指数表示について：取扱説明書 P.30 参照
指数表示では、数値を仮数部と指数部により表現します。
 $(2.85 \times 10^3)^2 + (62.98 \times 10^2)^2$ の計算は、 $(2.85 \times 1000)^2 + (62.98 \times 100)^2$ 、あるいは、 $(2850)^2 + (6298)^2$ と同じ意味です。
- 計算の優先順位について：取扱説明書 P.95 参照
優先順位は、
 - (1) π や変数の呼出し
 - (2) 関数
 - (3) べき乗
 - (4) 符号
 - (5) 乗除算
 - (6) 整数の除算
 - (7) 整数の乗除 (MOD)
 - (8) 加減算
 - (9) 大小比較
 - (10) 論理演算の順です（順位の若い方が優先されます）。

今日の練習問題

(問題 1) 以下の (1) から (10) に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) 9 + 8 \times 7 + 6 \times 5 + 4 \times 3 + 2 \times 1 =$$

$$(2) \frac{5.28 \times 3.33}{6.13 + 2.13 - 2.62} =$$

$$(3) 8.46 - \frac{8.32}{4.15 + 3.07 - 2.11} =$$

$$(4) \frac{5.23}{25 \times 6.22} + \frac{6.8}{333 \times 6.1} =$$

$$(5) (10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1)^{10} =$$

$$(6) 51 + 42^2 + 33^3 + 24^4 + 15^5 =$$

$$(7) \frac{2.2^{-5} + 3.4^{-2}}{5.4^{-4}} =$$

$$(8) (18 + 7)^3 \times 0.5 =$$

$$(9) \frac{1}{(87.93 + 24.15) \times (13.84 - 27.65)} =$$

$$(10) 4^3 + 7^2 + 8^4 =$$

(問題2) 以下の(1)から(10)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \sqrt{3+4} + \sqrt{5+6} =$$

$$(2) \sqrt{15(15-10)(15-5)(15-1)} =$$

$$(3) \sqrt{\frac{940}{8} \left(\frac{4.6 \times 10^5 \times 3.2 \times 10^5}{\pi^2} \right)^2} =$$

$$(4) \sqrt{34} \times \sqrt{86} =$$

$$(5) \sqrt{53+95} + \sqrt{0.84} =$$

$$(6) \ 18 \times \pi \times \sqrt{\frac{843}{257}} =$$

$$(7) \ \sqrt{12(12-5)(12-4)(12-3)} =$$

$$(8) \ \left(\frac{5.8 \times 10^2 + 3.8}{7.2 \times \sqrt{105}}\right)^3 =$$

$$(9) \ (3.2 + \sqrt{26.3 \times 8.1})^3 =$$

$$(10) \ \sqrt[3]{53 \times 0.25 + 72 \times 1.92} =$$

3. 算術代入計算の練習

算術代入計算について：取扱説明書 P.69 参照

複雑な計算を行う際には、「変数」を利用すると効率的です。PC-G850 シリーズの変数では、「英文字」「英文字+英文字」「英文字+数字」の3種類の組み合わせが利用できます。変数の長さは最大2文字までであることに注意して下さい（この制限はこのポケコン特有のものです）。また、変数の大文字と小文字は区別されないことにも注意が必要です。以下の例題を変数を用いて解いてみましょう。

（例題1）円錐振り子の周期 T は、糸の長さを L 、糸の鉛直となす角を θ とすると、

$$T = 2\pi \sqrt{\frac{L \cos \theta}{g}}$$

で求めることができる。 $L = 50[cm]$ 、 $\theta = 25^\circ$ 、 $g = 9.8[m/sec^2]$ の場合の周期 T を求めよ。

（例題2）月の軌道を、地球を中心とする円とみなし、地球の半径 r 、月の周期を T とすると、月の軌道半径は

$$R = \sqrt[3]{\frac{gr^2T^2}{4\pi^2}}$$

で与えることが出来る。地球の半径 $r = 6.4 \times 10^6[m]$ 、月の周期 $T = 27$ 日 8 時間としたときの月の軌道半径を求めよ。

（例題3）RL直列回路に流れる電流は、スイッチSWを閉じてから t 秒後に、

$$i(t) = \frac{E}{R} \left\{ 1 - \exp\left(-\frac{R}{L}t\right) \right\}$$

となる。今、 $E = 4.5[V]$ 、 $L = 160[mH]$ 、 $R = 55[\Omega]$ として、 $t = 0.1[ms]$ における電流の値を求めよ。

今日の練習問題

(問題1) 以下の(1)から(5)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \log 15 =$$

$$(2) \log 18.465 =$$

$$(3) \log 0.0018465 =$$

$$(4) \log 1.84 \times 10^{20} =$$

$$(5) 3.8 \log 0.49 =$$

(問題2) 以下の(1)から(5)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) e^{3.8} =$$

$$(2) e^{2 \times (3+7+5)} =$$

$$(3) 3.5 \times e^{-\frac{2}{7}} =$$

$$(4) \frac{19.3}{4.8+12.5 \times e^{-3.8}} =$$

$$(5) \frac{15.2}{2.3+8.5 \times e^{-4}} =$$

(問題3) 以下の(1)から(10)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \cos^2 30^\circ + \sin^2 30^\circ =$$

$$(2) \frac{1}{2} \times 18 \times 15.5 \times \sin 30^\circ =$$

$$(3) \tan^{-1} \frac{3 \sin 60^\circ}{4 + 3 \cos 60^\circ} =$$

$$(4) \cos(0.7\pi) =$$

$$(5) \frac{\cos 1.286}{\tan 6.254} =$$

$$(6) 4 \cos \frac{40^\circ}{2} \times \cos \frac{60^\circ}{2} \times \cos \frac{80^\circ}{2} =$$

$$(7) \cos^{-1} 0.345 =$$

4. 統計計算の練習

[ミニQ&A]

(Q) 変数の内容は電源を切って再び入れるとどうなる？

(A) そのまま記憶されています。

(Q) 数値の表現で用いる”E” と関数”EXP” の違いは何ですか？

(A) 例えば、数値 64000 を指数表示すると、 6.4×10^4 となり、この表現をポケコン上では、 $6.4E4$ と表現しています。他方、”EXP” は、説明書 P.39 にもあるように「指数関数（さらに言うと自然指数関数）」を指しています。以上が、数値表現のための記号”E” と関数”EXP” との違いです。

【利息計算その 1】 10 万円を年 6.7% で 5 年間預金をする、 $100000 \times \left(1 + \frac{6.7}{100}\right)^5$ になります。では、1 万円を年 0.2% で 3 年間預金をする、利息はいくらになりますか？

【利息計算その 2】 年利率 8% のときは、月利率は、 $\left(1 + \frac{8}{100}\right)^{\frac{1}{12}} - 1$ となります。では、年利率 15% のときは、月利率はいくらになりますか？

【統計計算に挑戦】 表 4.1 は、学生の一カ月当たりの親からもらう小遣いの調査結果です。小遣いの平均値と標準偏差を求めてみましょう。

表 4.1 学生の小遣い調査結果（自己申告に基づく）

金額	人数	金額	人数
500	3	3000	8
1000	4	3500	15
1500	6	4000	6
2000	7	4500	5
2500	10	5000	3

計算例：合計人数 n と小遣い合計 sm をまず求めます。

$$n = 3 + 4 + 6 + 7 + \cdots + 3$$

$$sm = 500 \times 3 + 1000 \times 4 + 1500 \times 6 + 2000 \times 7 + 2500 \times 10 + \cdots + 5000 \times 3$$

これより、小遣いの平均は $\frac{sm}{n}$ により求めることができます。標準偏差については、P.62 を参照してください（この例題は、ポケコン附属のアプリケーションプログラムを利用して解くことが出来ます P.64 の例題参照）。

（問題 1）以下の（1）から（2）に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \frac{6.2^{-3} + 3.3^{-6}}{4.5^{-5}} =$$

$$(2) \frac{3 \times 0.04 \times 10^{-2}}{4 + \frac{1}{2}} =$$

（問題 2）以下の（1）から（2）に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \sqrt[3]{53 \times 0.25 + 25 \times 12.92} =$$

$$(2) \sqrt[3]{3.2 + \sqrt{34.3 \times 2.1}} =$$

（問題 3）以下の（1）から（2）に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) 5.42 \log 5.44 =$$

$$(2) \log 321 =$$

（問題 4）以下の（1）から（2）に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) e^{3.8} =$$

$$(2) e^{7 \times (3+7+6)} =$$

（問題 5）以下の（1）から（2）に示す計算式をポケットコンピュータを用いて解きなさい。

$$[\text{RAD}] \quad (1) \sin 0.321 + 0.67 \times \cos(-0.126) - 4.55 \times \tan 0.95 =$$

$$[\text{DEG}] \quad (2) \tan^{-1} \frac{3 \sin 40^\circ}{65 + 35 \cos 50^\circ} =$$

（注）解答は、画面に表示される数値の全桁を記入すること。また、DIGIT 指定や USING 命令によるフォーマット指定は、全て解除しておくこと。

5. プログラミングの基礎

5.1 BASIC言語によるプログラミングの基礎

5.1.1 BASICとは何か？

BASICは、Beginner's All-purpose Symbolic Instruction Codeの略です。言い替えると、「初心者向けのあらゆる目的に合う記号を用いた命令語集合によるコンピュータ言語」となります。すなわち、BASIC言語は、コンピュータのための言語であると同時に、人間がプログラムを作成するための言語です。コンピュータのための言語は多数存在しています。例えば、PC-G850シリーズでは、BASIC言語のみならず、C言語、CASL（キャッスル）言語、アセンブリ言語、機械語の5種類の言語に対応しています。

5.1.2 プログラムとは何か？

（コンピュータ）プログラムとは、電子計算機が命令を実行する際の手順／手続きを記述したもので、比喩的に言うと、「入学式や卒業式のプログラム」や「運動会などの行事（イベント）のプログラム」と基本的に同じ意味を持ちます。プログラムは、文法および命令文や予約語などにより記述されます。プログラムを実際に作成する際には、決められた文法に従いながら、あらかじめ用意されている命令文を並べたり、組み合わせたりします。

5.1.3 プログラミングの基本はどう習得するのか？

全くの初心者には「コンピュータプログラム」についての概念が存在していません。従って、初心者はまず、「プログラミングを体験してみる」ことが、大切です。当然、何の準備や基礎知識もなしにプログラムを作成することは不可能です。この場合、他人のプログラムあるいは、書籍などに掲載されているやさしいプログラムを真似することから始めるのが近道です。初心者である間は、出来る限り短くまとまっていて、かつ、基本的なプログラムを実際にコンピュータに入力し実行する、そしてその実行結果を確認する、といった過程を繰り返します。この際に、自分の思いついたアイデアをプログラムに加えたり、プログラムの変更がどのような結果をもたらすのかを予測する訓練をすればさらに上達は加速されます。また、本当の意味でのプログラミング技能は、少なくとも数年間の訓

練習期間を経なければ身に付きません。ですから、「ゲームを作りたい」とか「ロボットをコンピュータで制御してみたい」といったような自分を励ます動機を持つことが何よりも大切です。

5.2 BASICプログラムを実行してみよう

以下に示すプログラムをポケコンに入力して実行してみてください。プログラムの入力方法と実行方法については、テキストの P.98 ～ P.105 に説明がありますので、分からなくなった時、参照してください。

5.2.1 画面に文字列を表示するプログラム

```
10 PRINT 'Hello, world!'  
20 END
```

注 1 : PRINT 文に関しては、P.338 を参照のこと。

注 2 : END 文に関しては、P.314 を参照のこと。

5.2.2 2変数の足し算を行うプログラム

```
10 INPUT A  
20 INPUT B  
30 C=A+B  
40 PRINT C  
50 END
```

注 3 : INPUT 文に関しては、P.324 を参照のこと。

5.2.3 足し算練習プログラム

```
10 INPUT 'A=';A  
20 INPUT 'B=';B  
30 C=A+B  
40 INPUT 'A+B=';D  
50 IF D=C THEN PRINT 'GOOD!' ELSE PRINT 'WRONG! Do it again!':GOTO 40  
60 PRINT 'Programming is fun!'  
70 END
```

注 4 : IF ～ THEN 文法に関しては、P.322 を参照のこと。

注 5 : GOTO 文に関しては、P.318 を参照のこと。

5.2.4 円の面積を計算するプログラム

```
10 INPUT R
20 S=PI*R^2
30 PRINT S
40 END
```

5.2.5 お楽しみプログラム（その 1）

```
10 CLS
20 WAIT 20:PRINT 'Hello!':WAIT 0
30 FOR I=-20 TO 170
40 GCURSOR (I,SIN(I*3)*20+28)
50 GPRINT '3C42ADC1C1AD423C'
60 GCURSOR (I-1,SIN(I*3)*20+28)
70 GPRINT '0000000000000000'
80 NEXT I
90 CLS:PRINT 'Bye-bye...'
```

注 6 : WAIT 文に関しては、P.350 を参照のこと。

注 7 : FOR ~ NEXT 文法に関しては、P.316 を参照のこと。

注 8 : GCURSOR 文に関しては、P.317 を参照のこと。

注 9 : GPRINT 文に関しては、P.318 を参照のこと。

注 10 : CLS 文に関しては、P.310 を参照のこと。

5.2.6 お楽しみプログラム（その 2）

```
10 CLS:DIM F(4)
20 X=0:F(0)=1:F(1)=4:F(2)=2:F(3)=3
30 WAIT 40:LOCATE 10,2:PRINT 'Wheel!':WAIT 0
40 FOR I=10 TO 150 STEP 10
50 CIRCLE(I,24),10,0,360,1,S,F(X)
60 CLS
70 X=X+1:IF X=4 THEN X=0
80 NEXT I
```

90 END

注 1 1 : CIRCLE 文に関しては、P.307 を参照のこと。

5.2.7 お楽しみプログラム（その 3）

```
10 RANDOMIZE
20 SC=0:REM スコアヨウノヘンスウ
30 V=1:REM イドウソクド
40 CLS
50 P=0:REM ヒョウジイチ
60 C$=CHR$( 65+RND26-1 ):REM ターゲットノモジ
70 LOCATE 0,0:PRINT "###タイピングレンシュウ ゲーム###"
80 LOCATE 0,5:PRINT "SCORE=";
90 PRINT USING "#####";SC;
100 LOCATE 12,5:PRINT "SPEED=";V;
110 FOR I=0 TO 23
120 LOCATE I,3:PRINT ">";
130 NEXT I
140 LOCATE 23,2:PRINT CHR$(233):REM ハートマーク
150 LOCATE P,2:PRINT C$
160 IF P=23 THEN LOCATE 9,2:PRINT "GAME OVER":END
170 I=1
180 A$=INKEY$:I=I+1
190 IF A$="" AND I<30 THEN 180
200 IF A$=C$ THEN WAIT 40:LOCATE 10,1:PRINT "HIT!":WAIT 0
    :SC=SC+(23-P):V=INT(SC/100)+1:GOTO 40
210 IF A$<>"" AND A$<>C$ THEN WAIT 40:LOCATE 9,1:PRINT
    "MISSED!":WAIT 0:SC=SC-(23-P):GOTO 40
220 LOCATE P,2:PRINT " "
230 P=P+V
240 IF P>23 THEN P=23
250 GOTO 150
```

6. BASIC言語でプログラミング

6.1 BASICで統計計算をするプログラムを作ろう

(例題) ある試験の点数をランダムに選出した35人について見た場合、表6.1のような結果が得られた。これより、平均値、標準偏差を求めよ。

表 6.1 試験の採点結果

点数	人数	点数	人数
30	1	70	8
40	1	80	9
50	4	90	5
60	5	100	2

(解答) 平均値 (av:average) と標準偏差 (sd:standard deviation) は以下の式により求めることができます。

$$av = \frac{\sum_{i=1}^N x_i}{N} \quad (6.1)$$

$$sd = \sqrt{\frac{\sum_{i=1}^N x_i^2 - N\bar{x}^2}{N-1}} \quad (6.2)$$

この例題を解くためのBASICプログラムを作成すると以下のようになります。

(参考) $av \simeq 71.43$, $sd \simeq 16.48$

```
10_S1=0:S2=0
20_INPUT_''サンプルスウ_N='';N
30_FOR_I=1_TO_N
40_PRINT_''データ'';I;''='';
50_INPUT_X
60_S1=S1+X:S2=S2+X^2
70_NEXT_I
80_AV=S1/N
90_PRINT_''ヘイキン_='';AV
100_SD=SQR((S2-N*AV^2)/(N-1))
110_PRINT_''ヒョウジュンヘンサ_='';SD
120_END
```

6.2 BASICでキャラクタを表示しよう

(例題) PC-G850 シリーズで表示できるキャラクタを全て画面に表示しなさい。

(解答) この例題を解くためのBASICプログラムを作成すると以下ようになります。キャラクターとそのコードの対応をまとめた表がテキスト P.370 にありますので参照してください。

```
10_FOR_I=32_TO_255
20_PRINT_CHR$(I);
```

```
30_NEXT_I
```

```
40_END
```

なお、キャラクタコードとそのキャラクタを一緒に表示する場合は以下のようになります（以下のプログラムでは、何かキーを押すと順次表示されて行きます）。

```
10_FOR_I=32_TO_255
```

```
20_PRINT_''CODE='';I;''_CHARACTER=[_];CHR$(I);''_]
```

```
30_IF_INKEY$='''_THEN_30
```

```
40_NEXT_I
```

```
50_END
```

6.3 BASICで数当てゲームを作ろう

（例題） 1 ～ 1 0 0 までの乱数を 3 回以内に当てるゲームを作成しなさい。

（解答） この例題を解くための BASIC プログラムを作成すると以下のようになります。

```
10_RANDOMIZE
```

```
20_PRINT_''GUESS_AND_HIT_THE_SECRET_NUMBER_Q.''
```

```
30_Q=RND_100
```

```
40_FOR_I=1_TO_3
```

```
50_INPUT_''YOUR_NUMBER='';A
```

```
60_IF_Q<A_THEN_PRINT_''IT_IS_BIGGER_THAN_Q.''
```

```

70 IF Q>A THEN PRINT 'IT IS SMALLER THAN Q.'
80 IF Q=A THEN PRINT 'YOU GOT THE NUMBER Q!':END
90 NEXT I
100 PRINT 'YOU LOST THE GAME.'
110 END

```

6.4 キャラクタを動かそう（その１）

（例題）キャラクタ“A”を画面の左端から右端まで移動させるプログラムを作成せよ。

（解答）この例題を解くためのBASICプログラムを作成すると以下のようになります。なお、PC-G850 シリーズの表示可能な座標範囲は、縦6行（0～5まで）、横24桁（0～23まで）です。

```

10 FOR I=0 TO 22
20 LOCATE I,3:PRINT 'A'
30 LOCATE I,3:PRINT ' '
40 NEXT I
50 END

```

上記プログラムを実行すると、キャラクタが高速に移動するため、見辛くなります。そこで、表示時間を調整するための空ループを挿入することによりこの問題を解決します。25行に以下の文を挿入してみましょう（この場合、多重ループになることに注意）。

```

25 FOR I2=1 TO 100:NEXT I2

```

多重ループを利用すると九九計算プログラムなどを簡単に作成することができます。

```

10 FOR I=1 TO 9
20 FOR J=1 TO 9
30 PRINT I;'*';J;'=';I*J
40 NEXT J
50 NEXT I

```

6.5 キャラクターを動かそう（その2）

（例題）キャラクター“A”を画面上で上下左右方向に移動させることのできるプログラムを作成せよ。

（解答）この例題を解くためのBASICプログラムを作成すると以下のようになります。

```
10 X=12:Y=3
20 LOCATE X,Y:PRINT 'A'
30 A$=INKEY$:IF A$=' ' THEN 20
40 LOCATE X,Y:PRINT ' '
50 IF A$='4' THEN X=X-1
60 IF A$='6' THEN X=X+1
70 IF A$='8' THEN Y=Y-1
80 IF A$='2' THEN Y=Y+1
90 GOTO 20
```

上記のプログラムを実行し、キャラクタを画面外に出そうとするとエラーが表示され、プログラムの実行が中断されてしまいます。画面外に出ないようにするには、50行から80行の文を以下のように変更します。

```
50 IF A$='4' AND X>0 THEN X=X-1
60 IF A$='6' AND X<23 THEN X=X+1
70 IF A$='8' AND Y>0 THEN Y=Y-1
80 IF A$='2' AND Y<5 THEN Y=Y+1
```

なお、上の50行から80行の文は、以下のような論理演算を利用することにより表現することも可能です（この表現を用いると、70行と80行は不要になります）。

```
50 X=X+(A$='4' AND X>0)-(A$='6' AND X<23)
60 Y=Y+(A$='8' AND Y>0)-(A$='2' AND Y<5)
```

さらに、上の50行と60行は以下のように表現することも可能です。

```
50 X=X-(A$='4')*(X>0)+(A$='6')*(X<23)
60 Y=Y-(A$='8')*(Y>0)+(A$='2')*(Y<5)
```

IF～THEN文では、～に相当する部分の式を論理演算し、その結果がゼロになるか、ゼロ以外になるかにより判定しています（コンピュータは最終的に二つ

の状態の判断しか行いません)。この考え方は、BASIC以外の言語でも同様ですので、論理演算の仕組みを是非理解してください。

なお、このプログラムを「お掃除ゲーム」にするには、25行に以下の文を挿入します。

```
25 LOCATE RND24-1,RND6-1:PRINT CHR$(RND64+33)
```

(考察) 論理演算を用いる場合と IF ～ THEN 文を利用する場合の実行速度を比較してみよう。

6.6 追いかけ(られる)ゲームを作ってみよう

(例題) 自分のキャラクタ“U”に向かって相手のキャラクタ“X”が追いかけてくるプログラムを作成せよ。ただし、捕まった場合の判定を行い、画面に“GAME OVER”と表示して終了すること。

(解答) この例題を解くためのBASICプログラムを作成すると以下のようになります。

```
10 X=0:Y=0:U=10:V=2:CT=0
20 LOCATE X,Y:PRINT 'U'
30 LOCATE RND24-1,RND6-1:PRINT '*'
40 GOSUB *TEKI1
50 A$=INKEY$:IF A$=' ' THEN 20
60 LOCATE X,Y:PRINT ' '
70 IF A$='4' AND X>0 THEN X=X-1
80 IF A$='6' AND X<23 THEN X=X+1
90 IF A$='8' AND Y>0 THEN Y=Y-1
100 IF A$='2' AND Y<5 THEN Y=Y+1
110 GOTO 20

120 REM テキノウゴキノショリ I
130 *TEKI1
140 CT=CT+1:IF CT>5 THEN CT=0 ELSE RETURN
150 LOCATE U,V:PRINT ' '
160 U=U-SGN(U-X)
170 V=V-SGN(V-Y)
180 D=SQR(SQU(U-X)+SQU(V-Y))
190 LOCATE 0,0:PRINT 'DISTANCE:';
```



```

200 PRINT USING '###';D
210 IF D>10 THEN GOSUB *WARP
220 IF X=U AND Y=V THEN LOCATE 9,2:PRINT 'GAME OVER':END
230 LOCATE U,V:PRINT 'X'
240 RETURN

250 REM ワープノショリ
260 *WARP
270 LOCATE 0,5:PRINT 'WARP!'
280 U=RND24-1
290 V=RND6-1
300 FOR I=1 TO 500:NEXT I
310 LOCATE 0,5:PRINT '      '
320 RETURN

330 REM テキノウゴキノショリ II
340 *TEKI2
350 LOCATE U,V:PRINT 'X'
360 B$=INKEY$:IF B$=' ' THEN RETURN
370 LOCATE U,V:PRINT ' '
380 IF A$='X' AND U>0 THEN U=U-1
390 IF A$='C' AND U<23 THEN U=U+1
400 IF A$='A' AND V>0 THEN V=V-1
410 IF A$='Z' AND V<5 THEN V=V+1
420 IF X=U AND Y=V THEN LOCATE 9,2:PRINT 'GAME OVER':END
430 LOCATE U,V:PRINT 'X'
440 RETURN

```

40行の“TEKI1”を“TEKI2”に変更すると、二人用の追いかけ（る／られる）ゲームになります（キーボードを壊さない様に）。

（考察1）相手の動きの速度を調整するには、どこ部分を調整すればよいか？

（考察2）相手を追いかけるプログラムにするには、どこ部分を調整すればよいか？また、その場合に何が問題となるかを考えよ。

6.7 追いかけゲームのプログラム

```
10 X=0:Y=0:U=10:V=2:CT=0
20 LOCATE X,Y:PRINT 'U'
30 LOCATE RND24-1,RND6-1:PRINT '*'
40 GOSUB *TEKI1
50 A$=INKEY$:IF A$=' ' THEN 20
60 LOCATE X,Y:PRINT ' '
70 IF A$='4' AND X>0 THEN X=X-1
80 IF A$='6' AND X<23 THEN X=X+1
90 IF A$='8' AND Y>0 THEN Y=Y-1
100 IF A$='2' AND Y<5 THEN Y=Y+1
110 GOTO 20

120 REM テキノウゴキノシヨリ I
130 *TEKI1
140 CT=CT+1:IF CT>5 THEN CT=0 ELSE RETURN
150 D=SQR(SQU(U-X)+SQU(V-Y))
160 LOCATE 0,0:PRINT 'DISTANCE:'
170 PRINT USING '###';D
180 IF D>8 THEN 270
190 LOCATE U,V:PRINT ' '
200 U=U+SGN(U-X)
210 V=V+SGN(V-Y)
```

```

220 IF U<0 THEN U=23
230 IF U>23 THEN U=0
240 IF V<0 THEN V=5
250 IF V>5 THEN V=0
260 IF RND10<5 THEN GOSUB *WARP
270 IF X=U AND Y=V THEN LOCATE 9,2:PRINT 'YOU WON!':END
280 LOCATE U,V:PRINT 'X'
290 RETURN

300 REM ワープノシヨリ
310 *WARP
320 LOCATE 0,5:PRINT 'WARP!'
330 U=RND24-1
340 V=RND6-1
350 FOR I=1 TO 500:NEXT I
360 LOCATE 0,5:PRINT '      '
370 RETURN

```

【ここで一言】

BASIC言語を利用したプログラミングがどのようなものか、少しずつ理解できるようになってきていると思います。BASICでは、基本的な命令語や文法を組み合わせることでプログラムを構成して行きます。追いかけてゲームプログラムの場合、これらの基本要素の理解だけでなく、プログラムの構成や手法（定石も含む）を理解する力が必要になります。前者の能力は比較的短期間で習得できますが、後者は経験を積んで行きながら獲得される能力です。ある問題を解くためのプログラムの表現方法は無数に存在し得るので、目的を達成するためにはどのように要素を構成すれば良いのかを発見したり見出したりするための試行錯誤・探索の能力が大切になります（この意味でプログラミングはパズルゲームに似ています）。

6.8 インベーダーゲームもどきを作ろう

これまでは2つのキャラクタを別々に移動させる処理を中心に説明してきました。本節では、アクションゲームなどで不可欠な玉の移動処理について焦点を当てます。より複雑な処理を記述する場合、プログラムは、メインとなる部分（**メインルーチン**）と必要時に呼び出すサブの部分（**サブルーチン**）に分けて作成します。この際、既に学習した **GOSUB ~ RETURN** 命令が役に立ちます。複雑なプログラムではバグを発見することが難しくなりますので、サブルーチン化により、プログラム全体の見通しや読みやすさに配慮しながらプログラミングすることが大切です。

6.8.1 初期化

以下の10～70行で、**変数の初期化**を行います。80行と90行では、ゲーム画面を描画します。DIM HX(2)と宣言することでHX(0),HX(1),HX(2)が使えるようになりますが、本プログラムでは、HX(1),HX(2)のみを使うため、HX(0)については初期化はしていません。なお、大砲（自分）の座標は(X,Y)で、インベーダ（箱型）の座標は(A,B)としています。玉の座標は(TX,TY)です。

```
10 DIM HX(2)
20 HX(1)=1:HX(2)=-1
30 H=1:SS=1:SC=0
40 CLS
50 A=52:B=0:X=72:Y=41
60 TX=0:TY=0:FLG=0
70 CT=0:SZ=10
80 LINE(40,0)-(40,47),S
90 LINE(102,0)-(102,47),S
```

6.8.2 各種情報の表示

以下の100～140行では、ラウンド数とスコアを画面に表示します。

```
100 REM MAIN ROUTINE
110 LOCATE 0,0:PRINT "ROUND:"
120 LOCATE 2,1:PRINT SS;
```

```
130 LOCATE 0,3:PRINT "SCORE:"
140 LOCATE 2,4:PRINT SC;
```

6.8.3 メインルーチン

以下の150行、160行では、大砲を描画します。180行では、インベーダの動きを処理するためにサブルーチン*TEKIを呼び出します。同様に190行では、玉を発射している場合（FLG=1）に限り、サブルーチン*MISSILEを呼び出します。200行では、インベーダが大砲と同じ縦座標に達したかどうかを判定し、そうであればゲーム終了処理へジャンプします。210行では、玉がインベーダに当たっているかどうかを判定し、もし当たっていれば、インベーダの爆発処理へジャンプします。220行～300行では、大砲の座標変更処理と再描画を行います。以下の150行から310行が本プログラムのメインルーチンになります。

```
150 LINE(X,Y)-(X+4,Y),S
160 LINE(X+2,Y)-(X+2,Y-2),S
170 C$=INKEY$
180 GOSUB *TEKI
190 IF FLG THEN GOSUB *MISSILE
200 IF B+SZ>=Y THEN *LOST
210 IF TX>=A AND TX<=A+SZ AND TY>=B AND TY<=B+SZ THEN *CLASH
220 IF C$="" THEN 170
230 VX=0
240 IF C$="4" AND POINT(X-1,Y)=0 THEN VX=-1
250 IF C$="6" AND POINT(X+5,Y)=0 THEN VX=1
260 IF C$=" " AND FLG=0 THEN FLG=1:TX=X+2:TY=Y-2
270 IF VX=0 THEN 170
280 LINE(X,Y)-(X+4,Y),R
290 LINE(X+2,Y)-(X+2,Y-2),R
300 X=X+VX
310 GOTO 150
```

6.8.4 玉の処理（サブルーチン）

以下の320～380行では、玉の移動と再描画処理を行います。玉の縦座標 TY が 0 になったとき、玉の処理を終了するため、変数 FLG の値を 0 に初期化します。

```
320 REM MOVE MISSILE
330 *MISSILE
340 PRESET(TX,TY)
350 TY=TY-1
360 IF TY<1 THEN FLG=0:RETURN
370 PSET(TX,TY)
380 RETURN
```

6.8.5 インベーダの処理（サブルーチン）

以下の390～470行では、インベーダの移動と再描画処理を行います。インベーダは端まで移動すると、5 ドット下に移動し、幅が 1 減少し、向きを変えて再度移動するようにします。なお、横に移動する速度は乱数により設定しています。

```
390 REM MOVE ENEMY
400 *TEKI
410 IF CT<3 THEN CT=CT+1:RETURN ELSE CT=0
420 LINE(A,B)-(A+SZ,B+SZ),R,&HFFFF,BF
430 A=A+HX(H)*RND(SZ*SS)
440 IF A+SZ>100 THEN B=B+5:H=2:SZ=SZ-1:A=100-SZ
450 IF A<44 THEN B=B+5:H=1:SZ=SZ-1:A=44
460 LINE(A,B)-(A+SZ,B+SZ),S,&HFFFF,BF
470 RETURN
```

6.8.6 インベーダの爆発処理（サブルーチン）

以下の480～560行では、インベーダの爆発処理を行います。玉の中心座標から徐々に大きな正方形を描画することで爆発しているような雰囲気を作ります。インベーダの爆破に伴ない、次のラウンドに進みます。また、スコアはインベーダの縦座標 B が大きい程、大きく加点されます。

```

480 REM HIT THE ENEMY
490 *CLASH
500 FOR SZ=1 TO 25*SS STEP 3
510 LINE(TX-SZ/2,TY-SZ/2)-(TX+SZ/2,TY+SZ/2),X,&HFFFF,B
520 NEXT SZ
530 LOCATE 6,2:PRINT "HIT!(";B;"Pts.)"
540 FOR I=1 TO 1000:NEXT I
550 SS=SS+1:SC=SC+B
560 GOTO 40

```

6.8.7 ゲーム終了処理（サブルーチン）

以下の570～630行では、ゲームの終了処理と再スタート処理を行います。大砲の中心座標から徐々に大きな正方形を描画することで爆発しているような雰囲気を作ります。最後の一言を表示した後、ユーザが何かキーを押すことにより、ゲームを再スタートさせるようにします。

```

570 REM SURRENDER
580 *LOST
590 FOR SZ=1 TO 100 STEP 3
600 LINE(X-SZ/2,Y-SZ/2)-(X+SZ/2,Y+SZ/2),X,&HFFFF,B
610 NEXT SZ
620 LOCATE 9,2:PRINT "Oh, NO!"
630 IF INKEY$="" THEN 630 ELSE 30

```

6.8.8 検討課題

- (1) インベーダの移動速度を変えるには、プログラムのどの部分を修正すれば良いか？
- (2) インベーダも大砲に向けて玉を発射できるようにするには、どうすれば良いか？
- (3) 複数の玉を発射できるようにするには、どうすれば良いか？

6.9 ダンジョン探検ゲームを作ろう

これまで学んできた BASIC 言語を用いて、ダンジョン探検ゲームを作成してみましょう。本節では、作成例のプログラムとその簡単な解説を示します。

6.9.1 ゲームのルール

ふとしたことからあなたは狭い狭いダンジョンに閉じ込められてしまいました。何とか出口を探して脱出しなければなりません。幸い脱出のためのアイテム（鍵）を持つ3人の商人がダンジョン内にはいるそうです。しかし、あなたの体力にも限界があり、一刻も早く脱出しなければなりません…。

6.9.2 ゲームの仕方

ダンジョン内には3種類の商人がいます。彼等を探し訪ねてください。このゲームをクリアするヒントは、本資料の中に隠されています。画面に表示されるダンジョン内を“2”, “4”, “6”, “8” キーで移動してください。残りのステップ数が0以下になるとゲームオーバーになります。

6.9.3 初期設定

```
10 CLS
```

```
20 DIM W$(9)*12, M$(1)*20
```

まず画面をクリアし、ダンジョンのマップ定義用配列を確保します。一般的な BASIC 言語では、DIM W\$(7) のみの宣言で済みますが、PC-G850 シリーズでは特に指定しない場合、DIM W\$(7) の宣言では、各要素 W\$(0), W\$(1), …, W\$(7) に格納可能な文字列の長さが7文字までとなってしまうため、本プログラムで必要な12文字分の文字列長を確保するために“*12”としています。

```
30 DATA "#####"
```

```
40 DATA "#####"
```

```
50 DATA "##.#G.Z##X##"
```

```
60 DATA "##.###.##.##"
```

```
70 DATA "##...##.##.##"
```

```
80 DATA "##.#.#.##.##"
```

```
90 DATA "##Y#.....##"
```

```
100 DATA "#####"
```

```
110 DATA "#####"
```

30行から110行までにダンジョン定義用データを記述しています（このマップは自由に変更できます）。

```
120 FOR I=0 TO 8
```

```
130 READ W$(I)
```

```
140 NEXT I
```

30行から始まるデータを配列 W\$() に格納します（**RUN の直後、RESTORE を実行したときと同じ状態に初期化される**ので、READ 命令で読み込まれるデータはプログラムの最も頭の方に存在するデータとなります。

```
150 LOCATE 5,0:PRINT "*** ダンジョン ***"
```

```
160 X=2:Y=2:FLG=0:P=60:GOSUB 310
```

ダンジョン内の主人公の位置座標を X,Y として初期座標を設定します。また、アイテムの獲得状態を表すフラグとして変数 FLG を 0 に初期化します。また、ダンジョン内を動き回る際の残りステップ数を 60 に設定します。その後、310行以降の**マップ表示のためのサブルーチン**を呼び出して初期画面を描画します。

6.9.4 メインプログラム

```
170 REM MAIN PROGRAM
```

```
180 A$=INKEY$:IF A$="" THEN 180
```

キーボードからの入力を受け付けます。何も入力されていない場合は、180行を繰り返し実行し、入力を待ちます。

```
190 VX=(A$="4")-(A$="6")-(A$="9")
```

```
200 VY=(A$="8")-(A$="2")+(A$="9")
```

キーボードからの入力に応じて、移動量を変数 VX と変数 VY に設定します。ここで、“9”を押した場合に、どのような移動量が設定されるのか考えてみましょう。また、プログラムを実行して実際にダンジョン内を探検した時にどのような役割があるのかについても考察してみましょう。

```
210 S$=MID$( W$(Y+VY),X+VX+1,1 )
```

```
220 IF S$="#" THEN VX=0:VY=0
```

「移動量が実際に変数 X と変数 Y に加算された状態」を想定し、移動後の位置にダンジョンの壁である“#”がないかどうかを調べます。もし壁が存在する場合、移動量を 0 に設定し、移動できないようにします。

```
230 IF S$="X" THEN IF FLG=0 THEN FLG=1:GOSUB *MESSAGE ELSE VX=0:VY=0:GOSUB *WHO
```

```
240 IF S$="Y" THEN IF FLG=1 THEN FLG=2:GOSUB *MESSAGE ELSE VX=0:VY=0:GOSUB *WHO
```

```
250 IF S$="Z" THEN IF FLG=2 THEN FLG=3:GOSUB *MESSAGE ELSE VX=0:VY=0:GOSUB *WHO
```

```
260 IF S$="G" THEN IF FLG=3 THEN *GOAL ELSE VX=0:VY=0:GOSUB *WHO
```

230 行から 260 行では、アイテムを取得する場合の判定処理や、ゴールする場合の判定処理などを行っています。具体的には、アイテム“X”を取得すると FLG=1 となり、アイテム“Y”を取得すると FLG=2 となり、さらにアイテム“Z”を取得すると FLG=3 となります。FLG=3 となった状態で“G”に接触するとゴールとなります。

```
270 X=X+VX:Y=Y+VY
```

```
280 IF VX OR VY THEN P=P-1:GOSUB 310
```

設定された移動量 VX と VY を変数 X と変数 Y にそれぞれ加算します。ここで、VX あるいは VY が 0 でない場合は移動することになりますので、画面を再描画するために 310 行以降のサブルーチンを呼び出します。

```
290 GOTO 180
```

メインルーチンの先頭に戻ります。

6.9.5 マップの表示

```
300 REM DISPLAY MAP
```

```
310 FOR I=-2 TO 2
```

```
320 FOR I2=-2 TO 2
```

```
330 LOCATE I2+5,I+3:PRINT MID$( W$(Y+I),X+I2+1,1 )
```

```
340 LOCATE 5,3:PRINT "A"
```

主人公の位置にキャラクタ“A”を表示します。

```

350 NEXT I2
360 NEXT I
370 LOCATE 10,1:PRINT "ITEMS: ";
380 IF FLG>0 THEN FOR I=1 TO FLG:PRINT CHR$(ASC("X")+I-1);" ";:NEXT I

```

取得したアイテムを表示します.

```

390 LOCATE 10,3:PRINT "STEPS: ";P

```

残りのステップ数を表示します.

```

400 IF P<=0 THEN *OVER

```

残りのステップ数が0以下になったらゲームオーバーとします.

```

410 RETURN

```

6.9.6 アイテム取得処理

```

420 REM DISPLAY MESSAGE
430 *MESSAGE
440 LOCATE 9,5:PRINT "ヨクキテクレタ!";CHR$(ASC("X")+FLG-1);"ヲアタエル"
450 FOR I=1 TO 1000:NEXT I
460 LOCATE 9,5:PRINT "                ":REM SPACEx15
470 RETURN

```

アイテムを取得した際のメッセージを画面に表示します.

6.9.7 拒絶処理

```

480 REM WHO ARE YOU?
490 *WHO
500 LOCATE 9,5:PRINT "オマエハナニモノジャ?"
510 FOR I=1 TO 1000:NEXT I
520 LOCATE 9,5:PRINT "チカラヲワケテモラウ!"
530 P=P-5
540 FOR I=1 TO 1000:NEXT I

```

```

550 LOCATE 9,5:PRINT "                                ":REM SPACEx13
560 RETURN

```

アイテムの準備不足で商人に拒絶される際のメッセージを画面に表示します.

6.9.8 ゴール処理

```

570 REM GOAL
580 *GOAL
590 CLS
600 M$(1)="オメデトウ！ ゴールデス！"
610 FOR I=1 TO LEN M$(1)
620 FOR I2=22 TO I+3 STEP -1
630 LOCATE I2,2:PRINT MID$( M$(1),I,1 );" "
640 FOR I3=1 TO 20:NEXT I3
650 NEXT I2
660 NEXT I
670 END

```

ゴールした際のメッセージを表示します（表示の仕方に注意してください）.

6.9.9 ゲームオーバー処理

```

680 REM GAME OVER
690 *OVER
700 CLS
710 M$(1)="ザンネン！ GAME OVER!"
720 FOR I=LEN M$(1) TO 1 STEP -1
730 FOR I2=1 TO 19+(I-LEN M$(1))
740 LOCATE I2,2:PRINT " ";MID$( M$(1),I,1 )
750 FOR I3=1 TO 20:NEXT I3
760 NEXT I2
770 NEXT I
780 END

```

ゲームオーバーの際のメッセージを表示します（表示の仕方に注意してください）。

【ここで一言】

今回の内容は難しく感じた人が多いかも知れません。その原因は、個々の命令が分からないというレベルではなく、「プログラムの構造やアルゴリズム」をまだ十分に修得していないことにあります。命令を英単語のように暗記しても「構造やアルゴリズム」の理解に到達することはありません。これらの能力はさまざまなプログラムを理解したり、接したり、作成してみて初めて身に付くものなのです。幸いなことに皆さんにはこれから存分にこうした能力を磨く時間も機会もあります。是非、プログラミングの魅力を理解して、コンピュータを身近な存在にしていってください。

6.10 ブロック崩しゲームを作ろう

これまで授業で学んできたBASIC言語を用いて、ブロック崩しゲームを作成してみましょう。本節では、作成例のプログラムとその簡単な解説を示します。

6.10.1 初期化

```
10 CLS
20 X=40:Y=24:L=5
30 A=RND(15)+60:B=RND(40)+5:H=1
40 FLG=0:S=0
50 DIM VX(9),VY(9)
60 RESTORE
```

10行から60行では、各種の初期化処理を行っています（10行：画面の初期化，20～50行：変数の初期化，60行：データ読み込み行の初期化）。なお、自分のラケットの座標をX，Yとし、その長さをLとします。また、ボールの座標をA，Bとし、進む方向をHとします。Sはスコア，FLGはボールの速度を調整するカウンタ変数とします。

6.10.2 移動量の設定

```
70 FOR I=1 TO 9
80 READ VX(I),VY(I)
90 NEXT I
100 DATA -1, 1, 0, 1, 1, 1
110 DATA -1, 0, 0, 0, 1, 0
120 DATA -1,-1, 0,-1, 1,-1
```

70行から120行では、ボールの進行方向の移動量を配列VXとVYに読み込みます。本プログラムでは、あらかじめ移動変化量を設定しておくことで、ボールを動かします。

6.10.3 ゲーム画面描画

```
130 LINE(40,0)-(104,0),S
140 LINE(40,47)-(104,47),S
150 LINE(104,0)-(104,47),S
```

```

160 FOR I=0 TO 4
170 FOR I2=0 TO 3
180 LINE(80+I2*5,5+I*8)-(80+I2*5+2,5+I*8+5),S,&HFFFF,BF
190 NEXT I2
200 NEXT I
210 LOCATE 0,0:PRINT "Score"

```

130行から210行までは、ゲーム画面の描画を行っています。LINE文により枠を描画し、さらにLINE文の塗りつぶし機能により、ブロックを描画します。さらにスコア表示位置にメッセージを表示します。

6.10.4 メインプログラム

```

220 REM MAIN PROGRAM
230 LINE(X,Y)-(X,Y+L),S
240 IF FLG=2 THEN FLG=0:GOSUB *BALL ELSE FLG=FLG+1
250 C$=INKEY$
260 IF C$="" THEN 230
270 LINE(X,Y)-(X,Y+L),R
280 IF C$="2" AND Y+L+1<47 THEN Y=Y+1
290 IF C$="8" AND Y-1>0 THEN Y=Y-1
300 GOTO 230

```

220行から300行までは、主に、自分が操作するラケットの移動処理を行います。キーボードからの入力を受け、それに応じてラケットの座標を変化させます。FLGが2になった時にサブルーチン*BALLが呼び出されることに注意してください。

6.10.5 ボールの移動

```

310 REM MOVE BALL
320 *BALL
330 PRESET(A,B)
340 IX=VX(H):IY=VY(H)
350 N=POINT(A+IX,B+IY)

```



```

360 IF N THEN 420
370 A=A+IX:B=B+IY
380 PSET(A,B)
390 IF A<35 THEN LOCATE 7,2:PRINT "GAME OVER!":END
400 RETURN

```

310行から400行では、ボールの移動処理を行っています。この際、移動先の画素情報をPOINT関数により検出し、障害物が存在する場合は、410行以降の処理に移り、障害物が存在しない場合は、ボールの座標値を変化させ、ゲームオーバーの判定を行い、メインプログラムに戻ります。

6.10.6 方向転換

```

410 REM CHANGE DIRECTION
420 IF POINT(A-1,B) AND POINT(A,B-1) AND H=7 THEN H=3:GOTO 580
430 IF POINT(A-1,B)=0 AND POINT(A,B-1)=0 AND H=7 THEN H=3:GOTO 580
440 IF POINT(A-1,B) AND POINT(A,B+1) AND H=1 THEN H=9:GOTO 580
450 IF POINT(A-1,B)=0 AND POINT(A,B+1)=0 AND H=1 THEN H=9:GOTO 580
460 IF POINT(A,B-1) AND POINT(A+1,B) AND H=9 THEN H=1:GOTO 580
470 IF POINT(A,B-1)=0 AND POINT(A+1,B)=0 AND H=9 THEN H=1:GOTO 580
480 IF POINT(A,B+1) AND POINT(A+1,B) AND H=3 THEN H=7:GOTO 580
490 IF POINT(A,B+1)=0 AND POINT(A+1,B)=0 AND H=3 THEN H=7:GOTO 580
500 IF POINT(A,B+1) AND H=3 THEN H=9:GOTO 580
510 IF POINT(A,B+1) AND H=1 THEN H=7:GOTO 580
520 IF POINT(A,B-1) AND H=9 THEN H=3:GOTO 580
530 IF POINT(A,B-1) AND H=7 THEN H=1:GOTO 580
540 IF POINT(A-1,B) AND H=1 THEN H=3:GOTO 580
550 IF POINT(A-1,B) AND H=7 THEN H=9:GOTO 580
560 IF POINT(A+1,B) AND H=3 THEN H=1:GOTO 580
570 IF POINT(A+1,B) AND H=9 THEN H=7
580 IF A+IX>79 AND A+IX<103 AND B+IY>2 AND B+IY<46 THEN PRESET(A+IX,B+IY):
    S=S+1:LOCATE 1,2:PRINT S
590 GOTO 340

```

410行から590行までは、ボールが障害物にぶつかった際の方向転換処理を行っています。このプログラムの中で最も注意すべき部分です。ボールの障害物へのぶつかり方が何通り存在するのかをまず考えます。そして、各場合での条件と転換する方向を明らかにして、IF文によりその内容を記述します。580行では、ボールが壁ではなく、ブロックにぶつかった時、衝突したブロックを消去して、スコアを1だけ加える処理をしています。

【ここで一言】

内容は理解できましたか？プログラミングの学習を始めたばかりの人なら、恐らく「難しい」と感じると思います。しかしながら、落胆する必要は全くありません。基本的に、他人の書いたプログラムを理解することは難しいものです。まず、はじめは、一生懸命真似をして、慣れてください。そして、その真似したプログラムをいじってみて、それがどのように影響するのかを体で覚えてください。自分でプログラムを作成してみたくなる日が必ずやってきます。もし、余裕のある人がいれば、このプログラムを二人用のブロック崩しゲームにしてみてください。その他にもいろいろ変更できます。

7. C言語でプログラミング（入門編）

7.1 C言語によるプログラミングの基礎

数多くのプログラミング言語に出会うことにより，プログラミングへの理解が深まるため，本授業では，これまでに取り上げてきたBASIC言語のみならず，C言語によるプログラミング方法について学習します．

7.2 足し算プログラムの比較（BASIC言語 VS C言語）

まず，以前に紹介した足し算プログラムについて，BASIC言語とC言語により作成した場合のプログラムを比較してみることになります．

【BASIC言語による場合】

```
10 PRINT "A=";  
20 INPUT A  
30 PRINT "B=";  
40 INPUT B  
50 C=A+B  
60 PRINT "A+B=";C  
70 END
```

【プログラムリスト1】

【C言語による場合】

```
10 main()  
20 {  
30     int a,b,c;  
40  
50     printf("A=");  
60     scanf("%d",&a);  
70     printf("B=");  
80     scanf("%d",&b);  
90     c=a+b;  
100    printf("A+B=%d ¥ n",c);  
110 }
```

【プログラムリスト2】

（注）通常，パソコンなどでC言語プログラムを作成する場合，行番号の入力は不要です．現在使用しているポケットコンピュータ PC-G850 シリーズでは，行番号を付ける〔約束〕になっています．

プログラムリスト1と2を比較してみると，次のことに気付きます．

☆C言語版では、アルファベットの小文字が使用されている：これは、C言語プログラムを記述する際の「ならわし」のようなものです。比較的初期のプログラミング言語（FORTRAN, COBOL, PL/I など）では、ディスプレイやプリンタなどの解像度を考慮して大文字をよく使用していました。C言語は、ディスプレイなどの解像度が飛躍的に進歩した時代に生まれましたので、これらのならわしに従う必要もなく、当然のこととして大文字と小文字の双方を使用できます。ただし、予約語などの基本的部分については、小文字が使用されるのが一般的です。

☆BASIC言語版の方が、プログラムが短い：一般的にBASIC言語版の方がプログラムが短くなる、とは言えません。実際に、プログラムリスト2の重要な部分は、50行から100行までの6行であり、BASIC版と同じ長さになっています。ただし、プログラム記述の様式（スタイル）の違いから、「やや」C言語のプログラムリストの方が長くなる傾向があります。

☆BASIC言語版には、main() などという表現は見当たらない：BASIC言語プログラムは、コマンド（命令文）により構成されています。C言語プログラムは、より広範な概念を扱うことのできる「関数 (functions)」により構成されます。main() 関数は、作成しようとしているプログラムそのものを関数により表現したものとみなすことが出来ます。C言語プログラムでは、まず main() 関数が存在することが第一条件なのです。「関数」はBASIC言語におけるコマンドとよく似ていますが、異なります。コマンドは「はじめから用意されていて、個々のプログラマが後になってその機能を追加・変更・更新することが基本的に困難」なものです。一方、「関数」は基本的に「始めから用意されておらず、プログラマがその機能を容易に追加・変更・更新できる」ものなのです。従って、この意味において「コマンド」と「関数」は対照的な存在なのです。

☆C言語版では知らない表現方法が多く使用されている：前節で説明しましたが、C言語は「関数」により記述されます。早速、main() 関数が登場しています。なお、一般に関数の書き方は以下のようになります。

関数の書き方：関数名（引数）

関数とその範囲の書き方：関数名（引数）{（範囲）}

また、C言語では「使用する変数は宣言する」約束になっています。このため、int a,b,c; という表現が出ていますが、これは、「整数型 (integer) の変数 a,b,c を確保しなさい」ということを意味しています。BASIC言語ではこのような宣言を記述することはありませんが、より正確にプログラマの

意図をコンピュータプログラムに反映させるには，こうした「型宣言 (type declaration)」が不可欠なのです．

7.3 C言語プログラミングの利点とは？

当然，どのようなプログラミング言語にも，プラス面とマイナス面があります．例えば，BASIC言語では，初心者に分かりやすいというプラス面がある一方で，インタプリタ方式を採用していることが多く，その結果，プログラムの実行速度が遅いというマイナス面があります．一方，C言語においては，BASIC言語ほど初心者向けではないというマイナス面がある一方で，コンパイラ方式を採用していることが多く，その結果，プログラムの実行速度が大変高速であるというプラス面があります．このように，コンピュータ言語は沢山存在していますが，それぞれに長所・短所があります．重要なことは，自分の目的を達成するのにどの言語を使用するのが最適なのかを見極めることのできる「目」を日頃から培うことであると思います．

7.4 参考図書

もし興味があれば，是非，本校の図書館や近所の書店などで以下に示すようなプログラミング言語の本を手にとってみてください．一冊お気に入りの本を購入しておくと，末永く利用できます．なお，高学年の授業で，年度始めの教科書購入の際に，プログラミング言語の本を購入する場合がありますので，急いで購入する必要はありません．

【是非一読を】 B. W. カーニハン／D. M. リッチー著，石田晴久訳：“プログラミング言語C ANSI規格準拠”，共立出版株式会社

【入門書】脇 英世著：“パソコンC言語入門－マイクロソフトCの実際－”，ブルーバックス B-729，講談社

【中級向け】三田典玄著：“実習C言語改訂新版”，アスキー出版局

【上級向け】Herbert Schildt 著，石田晴久監修，川崎盛美訳：“独習C++ Teach Yourself C++”，翔泳社，1994

7.5 BASICプログラミングとC言語プログラミングの違い

7.5.1 BASIC言語によるプログラミングでは

BASIC言語では、あらかじめ用意されているコマンドのみ（例えば、PRINT 命令、INPUT 命令など）を組み合わせることにより、プログラミングを行うのが普通です。一方、C言語ではコマンドではなく、関数によりプログラムを作成します。関数は、BASIC言語でのサブルーチンに近い性質を持っていますが、「再帰 (recursion) 呼び出し・処理」が可能であるなど、細かい点で異なります。また、関数により、プログラミングの複雑さは減少し、BASIC言語特有の「スパゲッティ」状態を容易に回避できるようになります。関数を利用しないBASIC言語プログラミングには、プログラムの保守や再利用が難しくなる弱点があります。

7.5.2 C言語によるプログラミングでは

もちろん、C言語においても、BASIC言語でよく利用する PRINT 命令や INPUT 命令に相当する関数があらかじめ【標準ライブラリ関数】として用意されています。従って、すべて一から関数を自分で作成する必要は全くありません。実際には、「足りない」「オリジナルな」機能に相当する関数を、その他の関数を利用して作成すれば良いだけなのです。さらに、C言語では、使用する変数を宣言することで、BASIC言語よりも厳密に変数の管理を行うため、変数使用の曖昧さによる危険を低減させています。以上に述べた点以外においても、柔軟性や拡張性、保守性、再利用性などにおいて、C言語はBASIC言語よりも優れた特徴を備えています。さて、C言語でプログラムを作成する際には、予約語 (reserved statement) と呼ばれるキーワードを用いて様々な処理を記述したり、変数などの宣言を行います。以下に予約語の一部を示します。

auto	double	int	struct	default	static
break	else	long	switch	goto	while
case	enum	register	typedef	sizeof	
char	extern	return	union	volatile	
const	float	short	unsigned	do	
continue	for	signed	void	if	

これらの予約語がどのような役割を果たすのかについて、具体的に以降のサンプルプログラムを見て行きましょう。

7.5.3 C言語プログラムの入力方法

これまでの授業で対象にしてきたBASICプログラムは、BASICエディタ（プログラムモード）を利用して入力・修正・保存・呼び出しなどを行ってききましたが、C言語プログラムは、PC-G850 シリーズに付属の【テキストエディタ】により入力・修正・保存・呼び出しなどを行います。テキストエディタの使用方は、取扱説明書 p.168 ～ p.183 で説明されていますので、熟読の上、修得してください。テキストエディタにより入力したプログラムを実行するには、まず、【コンパイル】という作業が必要になります。コンパイルの方法については、取扱説明書 p.186 ～ p.193 で説明されていますので、熟読の上、修得してください。コンパイル作業では、テキストエディタにより入力されたプログラムが、実行可能状態の機械語コードに変換されます。C言語プログラムが直接実行されるのではなく、生成された機械語コードが実行されます。

7.5.4 C言語プログラムを実行してみよう

以降に示すプログラムをポケコンに入力して実行してみてください。プログラムはBASIC版とC言語版を示していますので、両方を入力して実行してみてください。初歩の段階として、BASIC言語とC言語の違いを体感することは重要です。実行速度などについても比較してみると面白いでしょう。また、なぜBASIC版とC言語版で実行速度が異なるのか、考えてみてください。

画面に文字列を表示するプログラム

(BASIC版)

```
10 PRINT "Hello, World!"  
20 END
```

(C言語版)

```
10 main()  
20 {  
30   printf("Hello, World!¥ n");  
40   return 0;  
50 }
```

(注1) printf() (関数) に関しては、p.228 を参照のこと。

(注2) return (予約語) に関しては、p.206 を参照のこと。

円の面積を計算するプログラム

(BASIC版)

```
10 INPUT R
20 S = PI*R^2
30 PRINT S
40 END
```

(C言語版)

```
10 main()
20 {
30     double r,s;
40
50     scanf("%lf",&r);
60     s = 3.14*r*r;
70     printf("s=%f 円 n",s);
80     return 0;
90 }
```

(注3) scanf() (関数) に関しては, p.226 を参照のこと.

2変数の足し算を行うプログラム

(BASIC版)

```
10 INPUT A
20 INPUT B
30 C = A+B
40 PRINT C
50 END
```

(C言語版)

```
10 main()
20 {
30     int a,b,c;
40
```



```

50     scanf("%d",&a);
60     scanf("%d",&b);
70     c = a+b;
80     printf("a+b=%d ¥ n",c);
90     return 0;
100 }

```

足し算練習プログラム

(BASIC版)

```

10 INPUT "A=";A
20 INPUT "B=";B
30 C = A+B
40 INPUT "A+B=";D
50 IF D=C THEN PRINT "Good!" ELSE PRINT "Wrong!"
60 END

```

(C言語版)

```

10 main()
20 {
30     int a,b,c,d;
40
50     printf("A=");
60     scanf("%d",&a);
70     printf("B=");
80     scanf("%d",&b);
90     c = a+b;
100    printf("A+B=?");
110    scanf("%d",&d);
120    if ( d==c )
130    {
140        printf("Good! ¥ n");
150    } else {
160        printf("Wrong! ¥ n");
170    }
180    return 0;
190 }

```

(注4) if (予約語) に関しては, p.219 を参照のこと.

お楽しみプログラム (その1)

(BASIC版)

```
10 CLS
20 FOR I=-20 TO 170
30 GCURSOR (I,SIN(I*3)*20+28)
40 GPRINT "3C42ADC1C1AD423C"
50 NEXT I
60 END
```

(C言語版)

```
10 main()
20 {
30     int i;
40     double y;
50
60     clrscr();
70     for( i=-20; i<170 ;i++ )
80     {
90         y = sin((double)i*3.)*20.+28.;
100        gcursor( i,(int)y );
110        gprint("3C42ADC1C1AD423C");
120    }
130    return 0;
140 }
```

(注5) for (予約語) に関しては, p.202 を参照のこと.

(注6) gcursor() (関数) に関しては, p.237 を参照のこと.

(注7) gprint() (関数) に関しては, p.237 を参照のこと.

(注8) clrscr() (関数) に関しては, p.239 を参照のこと.

お楽しみプログラム（その2）

（BASIC版）

```
10 CLS:DIM F(4)
20 X=0:F(0)=1:F(1)=4:F(2)=2:F(3)=3
30 WAIT 40:LOCATE 10,2:PRINT "Wheel":WAIT 0
40 FOR I=10 TO 150 STEP 10
50 CIRCLE( I,24 ),10,0,360,1,S,F(X)
60 CLS
70 X=X+1:IF X=4 THEN X=0
80 NEXT I
90 END
```

（C言語版）

```
10 main()
20 {
30     int x=0,i;
40     int f[4]={1,4,2,3};
50
60     clrscr();
70     gotoxy( 10,2 );
80     printf("Wheel!");
90     for( i=10; i<150 ;i+=10 )
100    {
110        circle( i,24,10,0,360,1,0,f[x] );
120        clrscr();
130        if ( ++x == 4 ) x=0;
140    }
150    return 0;
160 }
```

（注9）circle()（関数）に関しては，p.237を参照のこと．

お楽しみプログラム（その3）

（C言語版のみ）

```

static char Map[20][20]={
    "000.....00000000000",
    "..0.....0.....0",
    "..0.0000.0.000000..0",
    "..0.0..0.0.0....0..0",
    "..000..0.0.0....0..0",
    ".....0.0.0000.0..0",
    "...00000.0....0.0..0",
    "...0.....0.0000.0..0",
    "...0..0000.0....0000",
    "...0..0....0.....",
    "0000..000..0000000..",
    "0.....0.....0..",
    "0000000.0.00000..0..",
    ".....0.0.0....0..0..",
    ".0000.0.0.0....0..0..",
    ".0..0.0.0.0....0000..",
    ".0..0.0.0.0.....",
    ".0..000.0.0.0000....",
    ".0.....0.0.0..0....",
    ".00000000.000..0000G"
};

int sc=0;

int DispMap( int cx, int cy )
{
    int i,i2,f;

    for( i=-2; i<3 ;++i )
        for( i2=-2; i2<3 ;++i2 )
        {
            f=( cx+i2<0 || cy+i<0 || cx+i2>19 || cy+i>19 );
            gotoxy( i2+5, i+3 );
            putchar( f==0 ? Map[cy+i][cx+i2] : '#' );
        }
    gotoxy( 5,3 );
}

```

```

    putchar('A');
    gotoxy( 12,0 );
    printf("Score: %d",++sc);
    /* getch(); */

    return 0;
}

int Goal()
{
    gotoxy( 9,2 );
    pritrnf("GOAL!!!");
    exit(0);
}

int Crash()
{
    gotoxy( 9,2 );
    pritrnf("Crash!!!");
    exit(0);
}

main()
{
    char c[9];
    int i,i2;
    int x=0, y=0, h=6;

    static struct DISPLACEMENT
    {
        int hx;
        int hy;
    } DM[10]={ { 0,0},
                { 0,0},{0, 1},{0,0},
                {-1,0},{0, 0},{1,0},
                { 0,0},{0,-1},{0,0}
    };

```

```

clrscr();

while( 1 )
{
    DispMap( x,y );

    c[0] = Map[y+DM[ h ].hy][x+DM[ h ].hx];
    c[2] = Map[y+1][x ];
    c[4] = Map[y ][x-1];
    c[6] = Map[y ][x+1];
    c[8] = Map[y-1][x ];

    if ( c[0]=='0' )
    {
        Map[y][x] = 'o';
        x += DM[ h ].hx;
        y += DM[ h ].hy;
    } else {
        /* Reached to Goal? */
        if ( c[0]=='G' ) Goal();

        /* Change Direction */
        for( i=2; i<9 ;i+=2 )
            if ( c[i]=='0' && h!=i)
            {
                h=i;
                break;
            }
        if ( i==10 ) Crash();
    }
}

return 0;
}

```

付録

A.【解答】 マニュアル計算の練習

- 指数表示について：取扱説明書 P.30 参照

指数表示では、数値を仮数部と指数部により表現します。

$(2.85 \times 10^3)^2 + (62.98 \times 10^2)^2$ の計算は、 $(2.85 \times 1000)^2 + (62.98 \times 100)^2$ 、あるいは、 $(2850)^2 + (6298)^2$ と同じ意味です。

- 計算の優先順位について：取扱説明書 P.95 参照

優先順位は以下の通りで、順位の若い方が優先されます。

- (1) π や変数の呼出し
- (2) 関数
- (3) べき乗
- (4) 符号
- (5) 乗除算
- (6) 整数の除算
- (7) 整数の乗除 (MOD)

- (8) 加減算
- (9) 大小比較
- (10) 論理演算

今日の練習問題

(問題1) 以下の(1)から(10)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) 9 + 8 \times 7 + 6 \times 5 + 4 \times 3 + 2 \times 1 = 109$$

$$(2) \frac{5.28 \times 3.33}{6.13 + 2.13 - 2.62} = 3.117446809$$

$$(3) 8.46 - \frac{8.32}{4.15 + 3.07 - 2.11} = 6.831819961$$

$$(4) \frac{5.23}{25 \times 6.22} + \frac{6.8}{333 \times 6.1} = 0.03698105$$

$$(5) (10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1)^{10} = 2.532951621E17$$

$$(6) 51 + 42^2 + 33^3 + 24^4 + 15^5 = 1128903$$

$$(7) \frac{2.2^{-5} + 3.4^{-2}}{5.4^{-4}} = 90.05500019$$

$$(8) (18 + 7)^3 \times 0.5 = 7812.5$$

$$(9) \frac{1}{(87.93 + 24.15) \times (13.84 - 27.65)} = -0.000646067$$

$$(10) 4^3 + 7^2 + 8^4 = 4209$$

(問題 2) 以下の (1) から (10) に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \sqrt{3+4} + \sqrt{5+6} = 5.962376101$$

$$(2) \sqrt{15(15-10)(15-5)(15-1)} = 102.4695077$$

$$(3) \sqrt{\frac{940}{8} \left(\frac{4.6 \times 10^5 \times 3.2 \times 10^5}{\pi^2} \right)^2} = 1.616690915E11$$

$$(4) \sqrt{34} \times \sqrt{86} = 54.07402334$$

$$(5) \sqrt{53+95} + \sqrt{0.84} = 13.0820402$$

$$(6) 18 \times \pi \times \sqrt{\frac{843}{257}} = 102.416423$$

$$(7) \sqrt{12(12-5)(12-4)(12-3)} = 77.76888838$$

$$(8) \left(\frac{5.8 \times 10^2 + 3.8}{7.2 \times \sqrt{105}} \right)^3 = 495.4625506$$

$$(9) (3.2 + \sqrt{26.3 \times 8.1})^3 = 5635.520646$$

$$(10) \sqrt[3]{53 \times 0.25 + 72 \times 1.92} = 5.330827816$$

B.【解答】算術代入計算の練習

算術代入計算について：取扱説明書 P.69 参照

複雑な計算を行う際には、「変数」を利用すると効率的です。PC-G850 シリーズの変数では、「英文字」「英文字+英文字」「英文字+数字」の3種類の組み合わせが利用できます。変数の長さは最大2文字までであることに注意して下さい（この制限はこのポケコン特有のものです）。また、変数の大文字と小文字は区別されないことにも注意が必要です。以下の例題を変数を用いて解いてみましょう。

（例題1）円錐振り子の周期 T は、糸の長さを L 、糸の鉛直となす角を θ とすると、

$$T = 2\pi \sqrt{\frac{L \cos \theta}{g}}$$

で求めることができる。 $L = 50[cm]$ 、 $\theta = 25^\circ$ 、 $g = 9.8[m/sec^2]$ の場合の周期 T を求めよ。

（例題2）月の軌道を、地球を中心とする円とみなし、地球の半径 r 、月の周期を T とすると、月の軌道半径は

$$R = \sqrt[3]{\frac{gr^2T^2}{4\pi^2}}$$

で与えることが出来る。地球の半径 $r = 6.4 \times 10^6[m]$ 、月の周期 $T = 27$ 日 8 時間としたときの月の軌道半径を求めよ。

（例題3）RL直列回路に流れる電流は、スイッチSWを閉じてから t 秒後に、

$$i(t) = \frac{E}{R} \left\{ 1 - \exp\left(-\frac{R}{L}t\right) \right\}$$

となる。今、 $E = 4.5[V]$ 、 $L = 160[mH]$ 、 $R = 55[\Omega]$ として、 $t = 0.1[ms]$ における電流の値を求めよ。

今日の練習問題

(問題1) 以下の(1)から(5)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \log 15 = 1.176091259$$

$$(2) \log 18.465 = 1.266349312$$

$$(3) \log 0.0018465 = -2.733650688$$

$$(4) \log 1.84 \times 10^{20} = 20.26481782$$

$$(5) 3.8 \log 0.49 = -1.177254896$$

(問題2) 以下の(1)から(5)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) e^{3.8} = 44.70118449$$

$$(2) e^{2 \times (3+7+5)} = 1.068647458E13$$

$$(3) 3.5 \times e^{-\frac{2}{7}} = 2.630170526$$

$$(4) \frac{19.3}{4.8+12.5 \times e^{-3.8}} = 3.799485856$$

$$(5) \frac{15.2}{2.3+8.5 \times e^{-4}} = 6.189724174$$

(問題3) 以下の(1)から(10)に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \cos^2 30^\circ + \sin^2 30^\circ = 1$$

$$(2) \frac{1}{2} \times 18 \times 15.5 \times \sin 30^\circ = 69.75$$

$$(3) \tan^{-1} \frac{3 \sin 60^\circ}{4 + 3 \cos 60^\circ} = 25.28499605$$

$$(4) \cos(0.7\pi) = -0.587785252$$

$$(5) \frac{\cos 1.286}{\tan 6.254} = -9.624096464$$

$$(6) 4 \cos \frac{40^\circ}{2} \times \cos \frac{60^\circ}{2} \times \cos \frac{80^\circ}{2} = 2.493620767$$

$$(7) \cos^{-1} 0.345 = 1.218557542$$

C.【解答】統計計算の練習

[ミニQ&A]

(Q) 変数の内容は電源を切って再び入れるとどうなる？

(A) そのまま記憶されています。

(Q) 数値の表現で用いる”E” と関数”EXP” の違いは何ですか？

(A) 例えば、数値 64000 を指数表示すると、 6.4×10^4 となり、この表現をポケコン上では、 $6.4E4$ と表現しています。他方、”EXP” は、説明書 P.39 にもあるように「指数関数（さらに言うと自然指数関数）」を指しています。以上が、数値表現のための記号”E” と関数”EXP” との違いです。

(利息計算その1) 10 万円を年 6.7% で 5 年間預金をすると、 $100000 \times \left(1 + \frac{6.7}{100}\right)^5$ になります。では、1 万円を年 0.2% で 3 年間預金をすると利息はいくらになりますか？

(5 年 : 138299.9736 3 年 : 10060.12008)

(利息計算その2) 年利 8% のときは、月利率は、 $\left(1 + \frac{8}{100}\right)^{\frac{1}{12}} - 1$ となります。では、年利 15% のときは、月利率はいくらになりますか？

(8 % : 0.00643403 15 % : 0.011714916)

(統計計算に挑戦) 表 C.1 は、学生の一カ月当たりの親からもらう小遣いの調査結果です。小遣いの平均値と標準偏差を求めてみましょう。

表 C.1 学生の小遣い調査結果（自己申告に基づく）

金額	人数	金額	人数
500	3	3000	8
1000	4	3500	15
1500	6	4000	6
2000	7	4500	5
2500	10	5000	3

計算例：合計人数 n と小遣い合計 sm をまず求めます。

$$n = 3 + 4 + 6 + 7 + \cdots + 3 = 67$$

$$sm = 500 \times 3 + 1000 \times 4 + 1500 \times 6 + 2000 \times 7 + 2500 \times 10 + \cdots + 5000 \times 3 = 191500$$

これより、小遣いの平均は $\frac{sm}{n} = 2858.208955$ により求めることができます。標準偏差については、P.62 を参照してください（この例題は、ポケコン附属のアプリケーションプログラムを利用して解くことが出来ます P.64 の例題参照）。

(問題 1) 以下の (1) から (2) に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \frac{6.2^{-3}+3.3^{-6}}{4.5^{-5}} = 9.171436708$$

$$(2) \frac{3 \times 0.04 \times 10^{-2}}{4 + \frac{1}{2}} = 0.000266666$$

(問題 2) 以下の (1) から (2) に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) \sqrt[3]{53 \times 0.25 + 25 \times 12.92} = 6.953777082$$

$$(2) \sqrt[3]{3.2 + \sqrt{34.3 \times 2.1}} = 2.269350727$$

(問題 3) 以下の (1) から (2) に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) 5.42 \log 5.44 = 3.986946036$$

$$(2) \log 321 = 2.506505032$$

(問題 4) 以下の (1) から (2) に示す計算式をポケットコンピュータを用いて解きなさい。

$$(1) e^{3.8} = 44.70118449$$

$$(2) e^{7 \times (3+7+6)} = 4.375039447E48$$

(問題 5) 以下の (1) から (2) に示す計算式をポケットコンピュータを用いて解きなさい。

$$[\text{RAD}] \quad (1) \sin 0.321 + 0.67 \times \cos(-0.126) - 4.55 \times \tan 0.95 = -5.38243657$$

$$[\text{DEG}] \quad (2) \tan^{-1} \frac{3 \sin 40^\circ}{65 + 35 \cos 50^\circ} = 1.262539894$$

(注) 解答は、画面に表示される数値の全桁を記入すること。また、DIGIT 指定や USING 命令によるフォーマット指定は、全て解除しておくこと。

D. 夏休みの課題

D.1 はじめに

本科目のシラバスにもありますように，9月以降の情報処理の授業では，BASIC言語を使ったプログラミングの話が本格化します．プログラミングの面白さを知ってもらうために，簡単なゲームの話もします．しかし，情報処理の講義時間は週1時間と少ないので，長めのプログラムについては，夏休みの課題として各自で入力しておいてもらうことにしました．以降に，プログラムリストを示しますので，夏休みの間にポケットコンピュータに入力すること．9月以降の授業では，すでにポケットコンピュータに以降のプログラムリストが保存されていることを前提として話を進めます．

（注意）プログラムは正確に入力すること．保存する際のファイル名は自由です．

D.2 プログラムリストその1

```
10 X=0:Y=0:U=10:V=2:CT=0
20 LOCATE X,Y:PRINT 'U'
30 LOCATE RND24-1,RND6-1:PRINT '*'
40 GOSUB *TEKI1
50 A$=INKEY$:IF A$=' ' THEN 20
60 LOCATE X,Y:PRINT ' '
70 IF A$='4' AND X>0 THEN X=X-1
80 IF A$='6' AND X<23 THEN X=X+1
90 IF A$='8' AND Y>0 THEN Y=Y-1
100 IF A$='2' AND Y<5 THEN Y=Y+1
110 GOTO 20
120 REM テキノウゴキノショリ I
130 *TEKI1
140 CT=CT+1:IF CT>5 THEN CT=0 ELSE RETURN
```

```

150 LOCATE U,V:PRINT ' '
160 U=U-SGN(U-X)
170 V=V-SGN(V-Y)
180 D=SQR(SQU(U-X)+SQU(V-Y))
190 LOCATE 0,0:PRINT 'DISTANCE: ';
200 PRINT USING '###';D
210 IF D>10 THEN GOSUB *WARP
220 IF X=U AND Y=V THEN LOCATE 9,2:PRINT 'GAME OVER':END
230 LOCATE U,V:PRINT 'X'
240 RETURN
250 REM ワープノシヨリ
260 *WARP
270 LOCATE 0,5:PRINT 'WARP!'
280 U=RND24-1
290 V=RND6-1
300 FOR I=1 TO 500:NEXT I
310 LOCATE 0,5:PRINT ' '
320 RETURN
330 REM テキノウゴキノシヨリ II
340 *TEKI2
350 LOCATE U,V:PRINT 'X'
360 B$=INKEY$:IF B$=' ' THEN RETURN
370 LOCATE U,V:PRINT ' '
380 IF A$='X' AND U>0 THEN U=U-1
390 IF A$='C' AND U<23 THEN U=U+1
400 IF A$='A' AND V>0 THEN V=V-1
410 IF A$='Z' AND V<5 THEN V=V+1
420 IF X=U AND Y=V THEN LOCATE 9,2:PRINT 'GAME OVER':END
430 LOCATE U,V:PRINT 'X'
440 RETURN

```


D.3 プログラムリストその2

```
10 X=0:Y=0:U=10:V=2:CT=0
20 LOCATE X,Y:PRINT 'U'
30 LOCATE RND24-1,RND6-1:PRINT '*'
40 GOSUB *TEKI1
50 A$=INKEY$:IF A$=' ' THEN 20
60 LOCATE X,Y:PRINT ' '
70 IF A$='4' AND X>0 THEN X=X-1
80 IF A$='6' AND X<23 THEN X=X+1
90 IF A$='8' AND Y>0 THEN Y=Y-1
100 IF A$='2' AND Y<5 THEN Y=Y+1
110 GOTO 20
120 REM テキノウゴキノショリ I
130 *TEKI1
140 CT=CT+1:IF CT>5 THEN CT=0 ELSE RETURN
150 D=SQR(SQU(U-X)+SQU(V-Y))
160 LOCATE 0,0:PRINT 'DISTANCE: ';
170 PRINT USING '###';D
180 IF D>8 THEN 270
190 LOCATE U,V:PRINT ' '
200 U=U+SGN(U-X)
210 V=V+SGN(V-Y)
220 IF U<0 THEN U=23
230 IF U>23 THEN U=0
240 IF V<0 THEN V=5
250 IF V>5 THEN V=0
260 IF RND10<5 THEN GOSUB *WARP
270 IF X=U AND Y=V THEN LOCATE 9,2:PRINT 'YOU WON!':END
280 LOCATE U,V:PRINT 'X'
290 RETURN
300 REM ワープノショリ
```

```

310_ WARP
320_ LOCATE_0,5:PRINT_ ' 'WARP! ' '
330_ U=RND24-1
340_ V=RND6-1
350_ FOR_ I=1_ TO_ 500:NEXT_ I
360_ LOCATE_0,5:PRINT_ ' ' ' ' ' ' ' '
370_ RETURN

```

D.4 プログラムリストその3

```

10_ DIM_ HX(2)
20_ HX(1)=1:HX(2)=-1
30_ H=1:SS=1:SC=0
40_ CLS
50_ A=52:B=0:X=72:Y=41
60_ TX=0:TY=0:FLG=0
70_ CT=0:SZ=10
80_ LINE(40,0)-(40,47),S
90_ LINE(102,0)-(102,47),S
100_ REM_ MAIN_ ROUTINE
110_ LOCATE_0,0:PRINT_ "ROUND:"
120_ LOCATE_2,1:PRINT_ SS;
130_ LOCATE_0,3:PRINT_ "SCORE:"
140_ LOCATE_2,4:PRINT_ SC;
150_ LINE(X,Y)-(X+4,Y),S
160_ LINE(X+2,Y)-(X+2,Y-2),S
170_ C$=INKEY$
180_ GOSUB_ *TEKI
190_ IF_ FLG_ THEN_ GOSUB_ *MISSILE
200_ IF_ B+SZ>=Y_ THEN_ *LOST
210_ IF_ TX>=A_ AND_ TX<=A+SZ_ AND_ TY>=B_ AND_ TY<=B+SZ_ THEN_ *CLASH
220_ IF_ C$=""_ THEN_ 170

```

```

230_VX=0
240_IF_C$="4"_AND_POINT(X-1,Y)=0_THEN_VX=-1
250_IF_C$="6"_AND_POINT(X+5,Y)=0_THEN_VX=1
260_IF_C$=""_AND_FLG=0_THEN_FLG=1:TX=X+2:TY=Y-2
270_IF_VX=0_THEN_170
280_LINE(X,Y)-(X+4,Y),R
290_LINE(X+2,Y)-(X+2,Y-2),R
300_X=X+VX
310_GOTO_150
320_REM_MOVE_MISSILE
330_*MISSILE
340_PRESET(TX,TY)
350_TY=TY-1
360_IF_TY<1_THEN_FLG=0:RETURN
370_PSET(TX,TY)
380_RETURN
390_REM_MOVE_ENEMY
400_*TEKI
410_IF_CT<3_THEN_CT=CT+1:RETURN_ELSE_CT=0
420_LINE(A,B)-(A+SZ,B+SZ),R,&HFFFF,BF
430_A=A+HX(H)*RND(SZ*SS)
440_IF_A+SZ>100_THEN_B=B+5:H=2:SZ=SZ-1:A=100-SZ
450_IF_A<44_THEN_B=B+5:H=1:SZ=SZ-1:A=44
460_LINE(A,B)-(A+SZ,B+SZ),S,&HFFFF,BF
470_RETURN
480_REM_HIT_THE_ENEMY
490_*CLASH
500_FOR_SZ=1_TO_25*SS_STEP_3
510_LINE(TX-SZ/2,TY-SZ/2)-(TX+SZ/2,TY+SZ/2),X,&HFFFF,B
520_NEXT_SZ
530_LOCATE_6,2:PRINT_"HIT!(";B;"Pts.*)"

```

```

540 FOR I=1 TO 1000: NEXT I
550 SS=SS+1: SC=SC+B
560 GOTO 40
570 REM SURRENDER
580 *LOST
590 FOR SZ=1 TO 100: STEP 3
600 LINE(X-SZ/2,Y-SZ/2)-(X+SZ/2,Y+SZ/2),X,&HFFFF,B
610 NEXT SZ
620 LOCATE 9,2: PRINT "Oh, NO!"
630 IF INKEY$="" THEN 630 ELSE 30

```

D.5 プログラムリストその4

```

10 CLS
20 DIM W$(9)*12,M$(1)*20
30 DATA "#####
40 DATA "#####
50 DATA "##. #G.Z##X##
60 DATA "##.###. #.##
70 DATA "##. .###.##
80 DATA "##. #. #. ##.##
90 DATA "##Y# . . . . ##
100 DATA "#####
110 DATA "#####
120 FOR I=0 TO 8
130 READ W$(I)
140 NEXT I
150 LOCATE 5,0: PRINT "***ダンジョン***"
160 X=2: Y=2: FLG=0: P=60: GOSUB 310
170 REM MAIN PROGRAM
180 A$=INKEY$: IF A$="" THEN 180
190 VX=(A$="4")-(A$="6")-(A$="9")

```

```

200_VY=(A$="8")-(A$="2")+(A$="9")
210_S$=MID$(W$(Y+VY),X+VX+1,1)
220_IF_S$="#"_THEN_VX=0:VY=0
230_IF_S$="X"_THEN_IF_FLG=0_THEN_FLG=1:GOSUB_*MESSAGE_ELSE_VX=0:VY=0:GOSUB_*WHO
240_IF_S$="Y"_THEN_IF_FLG=1_THEN_FLG=2:GOSUB_*MESSAGE_ELSE_VX=0:VY=0:GOSUB_*WHO
250_IF_S$="Z"_THEN_IF_FLG=2_THEN_FLG=3:GOSUB_*MESSAGE_ELSE_VX=0:VY=0:GOSUB_*WHO
260_IF_S$="G"_THEN_IF_FLG=3_THEN_*GOAL_ELSE_VX=0:VY=0:GOSUB_*WHO
270_X=X+VX:Y=Y+VY
280_IF_VX_OR_VY_THEN_P=P-1:GOSUB_310
290_GOTO_180
300_REM_DISPLAY_MAP
310_FOR_I=-2_TO_2
320_FOR_I2=-2_TO_2
330_LOCATE_I2+5,I+3:PRINT_MID$(W$(Y+I),X+I2+1,1)
340_LOCATE_5,3:PRINT_"A"
350_NEXT_I2
360_NEXT_I
370_LOCATE_10,1:PRINT_"ITEMS:";
380_IF_FLG>0_THEN_FOR_I=1_TO_FLG:PRINT_CHR$(ASC("X")+I-1);" ";:NEXT_I
390_LOCATE_10,3:PRINT_"STEPS:";P
400_IF_P<=0_THEN_*OVER
410_RETURN
420_REM_DISPLAY_MESSAGE
430_*MESSAGE
440_LOCATE_9,5:PRINT_"ヨクキテクレタ!";CHR$(ASC("X")+FLG-1);"ヲアタエル"
450_FOR_I=1_TO_1000:NEXT_I
460_LOCATE_9,5:PRINT_"aaaaaaaaaaaaaaaaaaaaaa":REM_SPACEx15
470_RETURN
480_REM_WHO_ARE_YOU?
490_*WHO
500_LOCATE_9,5:PRINT_"オマエハナニモノジャ?"

```

```

510 FOR I=1 TO 1000: NEXT I
520 LOCATE 9,5: PRINT "チカラヲワケデモラウ！"
530 P=P-5
540 FOR I=1 TO 1000: NEXT I
550 LOCATE 9,5: PRINT " ": REM SPACE*13
560 RETURN
570 REM GOAL
580 *GOAL
590 CLS
600 M$(1)="オメデトウ！ ゴールデス！"
610 FOR I=1 TO LEN M$(1)
620 FOR I2=22 TO I+3 STEP 1
630 LOCATE I2,2: PRINT MID$(M$(1),I,1); " "
640 FOR I3=1 TO 20: NEXT I3
650 NEXT I2
660 NEXT I
670 END
680 REM GAME OVER
690 *OVER
700 CLS
710 M$(1)="ザンネン！ GAME OVER!"
720 FOR I=LEN M$(1) TO 1 STEP -1
730 FOR I2=1 TO 19+(I-LEN M$(1))
740 LOCATE I2,2: PRINT " "; MID$(M$(1),I,1)
750 FOR I3=1 TO 20: NEXT I3
760 NEXT I2
770 NEXT I
780 END

```

D.6 プログラムリストその5

```

10 CLS

```

```

20_X=40:Y=24:L=5
30_A=RND(15)+60:B=RND(40)+5:H=1
40_FLG=0:S=0
50_DIM VX(9),VY(9)
60_RESTORE
70_FOR I=1 TO 9
80_READ VX(I),VY(I)
90_NEXT I
100_DATA -1,1,0,1,1,1
110_DATA -1,0,0,0,1,0
120_DATA -1,-1,0,-1,1,-1
130_LINE(40,0)-(104,0),S
140_LINE(40,47)-(104,47),S
150_LINE(104,0)-(104,47),S
160_FOR I=0 TO 4
170_FOR I2=0 TO 3
180_LINE(80+I2*5,5+I*8)-(80+I2*5+2,5+I*8+5),S,&HFFFF,BF
190_NEXT I2
200_NEXT I
210_LOCATE 0,0:PRINT "Score"
220_REM MAIN PROGRAM
230_LINE(X,Y)-(X,Y+L),S
240_IF FLG=2 THEN FLG=0:GOSUB *BALL ELSE FLG=FLG+1
250_C$=INKEY$
260_IF C$="" THEN 230
270_LINE(X,Y)-(X,Y+L),R
280_IF C$="2" AND Y+L+1<47 THEN Y=Y+1
290_IF C$="8" AND Y-1>0 THEN Y=Y-1
300_GOTO 230
310_REM MOVE BALL
320_*BALL

```

```

330 PRESET(A,B)
340 IX=VX(H):IY=VY(H)
350 N=POINT(A+IX,B+IY)
360 IF N THEN 420
370 A=A+IX:B=B+IY
380 PSET(A,B)
390 IF A<35 THEN LOCATE 7,2:PRINT "GAME OVER!":END
400 RETURN
410 REM CHANGE DIRECTION
420 IF POINT(A-1,B) AND POINT(A,B-1) AND H=7 THEN H=3:GOTO 580
430 IF POINT(A-1,B)=0 AND POINT(A,B-1)=0 AND H=7 THEN H=3:GOTO 580
440 IF POINT(A-1,B) AND POINT(A,B+1) AND H=1 THEN H=9:GOTO 580
450 IF POINT(A-1,B)=0 AND POINT(A,B+1)=0 AND H=1 THEN H=9:GOTO 580
460 IF POINT(A,B-1) AND POINT(A+1,B) AND H=9 THEN H=1:GOTO 580
470 IF POINT(A,B-1)=0 AND POINT(A+1,B)=0 AND H=9 THEN H=1:GOTO 580
480 IF POINT(A,B+1) AND POINT(A+1,B) AND H=3 THEN H=7:GOTO 580
490 IF POINT(A,B+1)=0 AND POINT(A+1,B)=0 AND H=3 THEN H=7:GOTO 580
500 IF POINT(A,B+1) AND H=3 THEN H=9:GOTO 580
510 IF POINT(A,B+1) AND H=1 THEN H=7:GOTO 580
520 IF POINT(A,B-1) AND H=9 THEN H=3:GOTO 580
530 IF POINT(A,B-1) AND H=7 THEN H=1:GOTO 580
540 IF POINT(A-1,B) AND H=1 THEN H=3:GOTO 580
550 IF POINT(A-1,B) AND H=7 THEN H=9:GOTO 580
560 IF POINT(A+1,B) AND H=3 THEN H=1:GOTO 580
570 IF POINT(A+1,B) AND H=9 THEN H=7
580 IF A+IX>79 AND A+IX<103 AND B+IY>2 AND B+IY<46 THEN PRESET(A+IX,B+IY):
    S=S+1:LOCATE 1,2:PRINT S
590 GOTO 340

```


【ここで一言】

ポケコンのキーボードに慣れて、スムーズに入力できましたか？ポケコンのキーボードのキー配列とパソコンのキーボードのキー配列がほぼ同じ（アルファベットについて）であることに気付きましたか？このようなキー配列のことを“QWERTY配列”と呼びます。なお、各プログラムリストでゲームを楽しめます。詳細は秋以降の授業で説明しますので楽しみにしてください。

E. 冬休みの課題

E.1 はじめに

夏休みの課題では，幾つかのゲームプログラムをポケコンに入力してもらいました．また，9月以降の授業では，各ゲームプログラムについて授業で説明しました．現在では，基本的なBASICプログラムを読めるようになっていることと思います．ただ，これまでは各自でプログラムを作ってもらったといった実習的なことをしていません．そこで，冬休みの余暇時間を利用して，各自で1つ，ゲームプログラムを作成してもらった課題を出したいと思います．以下の課題から1つ選んで各自プログラムを作成し，それをA4レポートとしてまとめた後，冬休み明けの情報処理の授業の際に提出すること．シラバスにもありますように，このレポートの評価は，最終成績の10％に反映させます．

【選択課題1】夏休みの課題で提示した「プログラムリストその1」を改良して**二人用の**追いかけゲームを作成しなさい．

【選択課題2】夏休みの課題で提示した「プログラムリストその3」を改良して**複数の玉を発射できる**インベーダゲームもどきを作成しなさい．

【選択課題3】夏休みの課題で提示した「プログラムリストその4」を改良して**地下2階まである**ダンジョンゲームを作成しなさい．

【選択課題4】夏休みの課題で提示した「プログラムリストその5」を改良して**2人用**テニスゲームを作成しなさい．

【選択課題 5】自分で自由課題（ゲーム）を設定し，そのプログラムを作成しなさい．なお，プログラムの長短は問いません．

【ここで一言】

この課題で初めて自分のアイデアをプログラムにする人が多いと思います．慣れない間は「難しい」と思うかも知れませんが，自分のアイデアがコンピュータ上で生き活きと動くというのは面白いと思いませんか．是非「自分で作ったゲームで遊ぶ」ことの面白さを実感してもらえればと思います．プログラミングは大変奥深いと同時に苦労が多い作業ですが，一度完成すれば，大きな達成感を得ることができます．なお，作成した自分のゲームプログラムは必ず保存しておいてください．冬休み明けの授業で，ポケコンを交換して遊ぶ時間を少し設ける予定です．