

教育利用を目的とした簡易な Basic のインタプリタ

Plain_Basic プログラミング例題

2006-08-20 版

目次

- C1. 数の計算
 - C1.0 一般
 - C1.1 べき乗計算
 - C1.2 三角関数の計算
 - C1.3 対数関数と指数関数
 - C1.4 切り上げ・切り捨て・四捨五入
- C2. 代数式の数値計算
 - C2.0 一般
 - C2.1 二次方程式の解
 - C2.2 三次方程式の解 (カルダノの解法)
 - C2.3 三次方程式の解 (ニュートン法)
 - C2.4 二元連立一次方程式を解く
- C3. グラフィックス
 - C3.0 一般
 - C3.1 多角形の作図
 - C3.2 円の作図で描く紋様
 - C3.3 乱数を応用した矩形の作図
 - C3.4 文字の作図例
- C4. 関数の作図
 - C4.0 一般
 - C4.1 三次方程式($Y=X^3-3X-4$)のグラフ
 - C4.2 楕円の作図
 - C4.3 イラストの作図

表 C1 drawpolyline.txt のリスト

表 C2 ex43panda.txt のリスト

表 C3 drawcurve.txt のリスト

C1. 数の計算

C1.0 一般

Plain_Basic は、普通の電卓なみに、数値を直接使った数値計算ができます。キー操作の方法が少し違います。電卓で単純な計算をさせるとき、例えば

$$3.15 \times 5 \div 7 =$$

と順にキーを押せば、答の 2.25 が得られます。Plain_Basic では下ののように頭にキーワードの PRINT をつけて入力します。キーボードには×÷記号がありませんので、代わりに * / を使います。

```
PRINT 3.15 * 5 / 7      (Enter↓)
```

Plain_Basic では、変数名を使った代数式を使うことができ、それらをプログラムに組み立てて保存し、何度でも計算ができます。以下の例題計算は、関数電卓でもできる数値計算ですが、プログラムにしていると、多様な応用を試すことができます。数の型は整数型と実数型を正しく使い分ける必要があります。前に変数名の型を指定し直した処理があると、型指定が正しく機能しないこともありますので、NEW コマンドで初期化しておいて、プログラムを LOAD するのが安全です。

C1.1 べき乗計算

ある数の 2 乗・3 乗・平方根・立方根を計算するのは、すべて、べき乗の演算子 \wedge を使って、2.3,(1/2),(1/3) 乗で計算できます。2 乗と 3 乗とは掛け算の繰り返しの方が単純です。平方根は、良く使うこともあって、関数 SQR を準備してあります。立方根は、専用の関数を特に準備しませんので、1/3 乗で計算します。例題プログラムは内部で A=2 と設定した計算です。任意の数を計算したいときには、直接モードで、例えば「A=5: GOTO 20」と入力します (図 1.1 ex11power.txt)。

C1.2 三角関数の計算

組み込みの三角関数 SIN, COS, TAN の計算に使う角度は、数学ではラジアンを引き数にするのですが、実用計算では度を使うのが分かり易くて便利です。Plain_Basic では、デフォルトとして、度を使うことにしました。計算の途中でラジアン単位を使いたいときは、コマンド「OPTION RADIAN」と入力します。度に直すときは、「OPTION DEGREE」と入力します。NEW コマンドは、Plain_Basic を初期化しますが、デフォルトの度の設定に戻ります。例題では、円周率 π の数値と、逆正接(ATN,ATN2)の計算をするときにラジアンに直して計算しています (図 1.2 ex12angle.txt)。

C1.3 対数関数と指数関数

底に 10 を使う対数が常用対数であって、組み込み関数 LOG を準備してあります。e を底に使うのは自然対数と言いますが、数学記号では $\ln(x)$ と表記します。この計算は常用対数を使って、 $\log(x)/\log(e)$ で計算できます。このとき e の値が必要になりますが、組み込みの指数関数を使って EXP(1) で求めることができます。対数計算とべき乗計算とは相互に逆関数の関係です。EXP は e のべき乗を計算する関数です。工学では常用対数に根拠を持つデシベル(db)と呼ぶ数を良く使います。これは、電圧や電流などの倍率 r を $20 \times \text{LOG}(r)$ で書き換えた値です。例題では 5 db が何倍に当たるかの計算も入れてあります。(図 1.3 ex13logarithm.txt)。

C1.4 切り上げ・切り捨て・四捨五入

実社会で扱う数は、すべて整数扱いです。小数点の付いた数は、小数点を飾りのように考えて、小数点以下の何桁目かに実質的な単位がある整数として計算します。少し高級な電卓では、切り上げ・切り捨て・四捨五入(up, cut, 5/4)の記号の付いた切り換えセレクトがあります。また、小数以下何桁目でこれを行わせるかのセレクトもあって(F.4.3.2.1.0)のどれかを選びます。このような数値の扱いを丸めと言います。お金の単位で、例えばドルはセント単位まであります。ドル単位で表すと小数二桁までの数ですが、二桁以上の表現は使いません。円とドルとの比率を扱うときは考え方が違います。現実のお金の計算では、端数を切り捨てます。衣料材料を準備するときの数値は、必要長さを切り上げた寸法数値の布材を使います。科学技術計算では、最低桁で四捨五入を使います。数の丸めは、実生活で習慣として処理していて意識していないことが多いのですが、数値計算のプログラミングでは、案外面倒なところがあります。例題では、整数型・実数型、整数除算、余りの計算を扱っています。小学校で習う「10 割る 3 は、答 3、余り 1」の計算は、丸めの概念を使っています。例題は、任意の小数を、小数点以下 3 桁で丸める計算を示します。(図 1.4 ex14rounding.txt)。

```

D:\borland\#PBasic\#ex11power.txt
list
10 A=2
20 PRINT "data A=";A
30 PRINT "平方 A**A=";A^2
40 PRINT "立方 A**A**A=";A**A**A
50 PRINT "平方根=";SQR(A)
60 PRINT "立方根=";A^(1/3)
run
data A= 2
平方 A**A= 4
立方 A**A**A= 8
平方根= 1.4142135623731
立方根= 1.25992104989487

```

图 1.1 ex11power.txt

```

D:\borland\#PBasic\#ex12angle.txt
list
10 ANG=30
20 PRINT "ANG="; ANG; " in degree"
30 S=SIN(ANG):PRINT "S=SIN(ANG)=";S
40 C=COS(ANG):PRINT "C=COS(ANG)=";C
50 T=TAN(ANG):PRINT "T=TAN(ANG)=";T
60 PRINT "ATN(T)=";ATN(T)
70 PRINT "ATN2(S,C)=";ATN2(S,C)
80 OPTION RADIAN
90 PRINT "angle T is in radian"
100 PRINT "ATN(T)=";ATN(T)
110 PRINT "PAI=4*ATN(1)=";4*ATN(1)
120 OPTION DEGREE
run
ANG= 30 in degree
S=SIN(ANG)= 0.5
C=COS(ANG)= 0.866025403784439
T=TAN(ANG)= 0.577350269189626
ATN(T)= 30
ATN2(S,C)= 30
angle T is in radian
ATN(T)= 0.523598775598299
PAI=4*ATN(1)= 3.14159265358979

```

图 1.2 ex12angle.txt

```

D:\borland\#PBasic\#ex13logarithm.txt
list
10 A=5
20 PRINT "A="; A
30 T=LOG(A):PRINT "T=LOG(A)="; T
40 PRINT "BASE e="; EXP(1)
50 PRINT "ln(A)=";LOG(A)/LOG(EXP(1))
60 PRINT "A db="; 10^(A/20);"倍です"
run
A= 5
BASE e= 2.71828182845905
ln(A)= 1.6094379124341
A db= 1.77827941003892 倍です

```

图 1.3 ex13logarithm.txt

```

D:\borland\#PBasic\#ex14rounding.txt
list
10 RA=10.12345: REM 被除数
20 RB=3.1416 : REM 除数
30 PRINT "被除数RA="; RA
40 PRINT "除数 RB="; RB
50 RC=RA/RB
60 IC=RC
70 PRINT "实数商 RC=RA/RB="; RC
80 PRINT "整数商 IC=RA/RB="; IC
90 PRINT "剩余=";MOD(RA,RB)
100 ID=1000*RC:RD=ID:RD=RD/1000
110 PRINT "小数点以下3桁で"
120 ID=1000*RC:RD=ID:RD=RD/1000
130 PRINT "切り捨て RC=";RD
140 IF RC>RD THEN RD=RD+1/1000
150 PRINT "切り上げ RC=";RD
160 ID=1000*(RC+0.0005):RD=ID:RD=RD/1000
170 PRINT "四捨五入 RC=";RD
run
被除数RA= 10.12345
除数 RB= 3.1416
实数商 RC=RA/RB= 3.22238668194551
整数商 IC=RA/RB= 3
剩余= 0.69865
小数点以下3桁で
切り捨て RC= 3.222
切り上げ RC= 3.223
四捨五入 RC= 3.222

```

图 1.4 ex14rounding.txt

C2. 代数式の数値計算

C2.0 一般

Plain_Basic は、文字と記号で表された代数式の数値計算をさせることができます。理屈は分かっていますが、具体的に数値計算で値を求めたいとなると、面倒で手間が掛かりことも少なくありません。ここで上げた4つの例題は、中学高校の数学で紹介されている簡単な問題です。ほぼ数学式の通りにプログラミングをすればよいのですが、Plain_Basic で決めた文法に合わせるための工夫が必要です。個々の例題は、コンソールにリスト表示された Basic 文のソースコードと実行結果をスクリーンコピーして表示しました。ソースコードの行数を節約するため、コロン(:)を使ってマルチステートメントを多用しています。例題プログラムは、使い方の説明をするため、組み込みの例題数値で実行させますが、ユーザのデータを使って GOTO 文で走らせる方法が分かるようにしてあります。この例題のような独立した小さなプログラム単位の作成は、大きなプログラムを作成するときに、サブルーチン単位の設計と実行テストとして応用されます。

C2.1 二次方程式の解

プログラミング流の書式で二次方程式を書くと、 $Y=A*X^2+B*X+C$ です。定数 A,B,C を与えて、 $Y=0$ を満たす解 $X1, X2$ を求めます。中学校数学問題の定番ですが、具体的に数値計算をするとなると、平方根の計算が入りますし、解が一意には求まらない、などの面倒さが入ります。プログラムでは、ラベル 200 と 300 に別の例題用数値を入れてあります。同じように、実用に使うときは、定数 A,B,C を直接モードで設定しておいて、GOTO 40 とすれば解が求まります。(図 2.1 ex21equation2.txt)。

C2.2 三次方程式の解 (カルダノの解法)

問題は、三次方程式は、 $Y=X^3+A*X^2+B*X+C=0$ を満たす X の解を計算することです。ここではカルダノ(Cardano,1501-1576)の公式を使います。具体的には、3次の対称マトリックスの固有値を計算するときに使えます。実用例は、鉄筋コンクリート梁の設計計算のときに現われます。公式は多くの参考書で紹介されていますが、実際に数値計算をするとなると手間が掛かりますので、あらかじめ簡単なプログラムが使えるようにしてあると便利です。なお、数学的には複素数の解もありますが、実数解だけを示しました。また、3つの実数解があるとき、大小順に並べ直しをしていません。

(図 2.2 ex22equation3.txt)。

C2.3 三次方程式の解 (ニュートン法)

ニュートン法で、何かの関数の逆関数を求める方法の説明用に作りました。上の例題は、上と同じ三次方程式です。一般的な問題の応用には、解きたい関数本体 Y 、その一回微分した式 dY を差し換えればよいのです。このプログラミングの場合、元の三次式を表す方法として、べき乗の演算子 $^$ を使わずに、多重括弧を使って $Y=(X+A)*X+B)*X+C$ の形に変形していることに注目して下さい。Newton 法では、初期値から出発して補正を繰り返して一つの解を得ます。二つ以上の解があるときは、あらかじめ、どのあたりに解があるかの初期値を入れ直さなければなりません。また、繰り返し計算を止める条件として、精度を決めておかなければなりません。もし極大または極小値があると無限ループに入る危険があります。これを避けるため、FOR-NEXT 文で 50 回の繰り返しで計算を停止するようにしてあります。

(図 2.3 ex23Newton.txt)。

C2.4 二元連立一次方程式を解く

二つの直線の交点を求める問題は、典型的な二元連立一次方程式を解く問題になります。代数式では連立一次方程式をマトリックスで表すことが多いのですが、プログラミングではマトリックス記号を理解する約束が使えませんので、単純な代数式の形で扱います。この例題では、6個の定数を入力データとしなければなりません。その入力方法に内部ファイルの読み込み(READ-DATA 文)を応用しています。ラベル番号 140 の DATA 文を差しかえると、別の計算ができます。ラベル 150 を付けた単独の DATA 文をプログラムファイルに作成しておいて、MERGE 文で読み込ませる方法が応用できます。また、直接モードで定数にデータを代入しておいて、GOTO 70 で走らせる方法も使うことができます。

(図 2.4 ex24lineareq2.txt)。

```

D:\borland\#PBasic\#ex21equation2.txt
list
10 REM "二次方程式 AX^2+BX+C=0 を解く"
20 REM "最初は例題です"
30 A=2: B=3: C=-3
40 D=B*B - 4*A*C
50 IF D<0 THEN GOTO 110
60 IF D=0 THEN GOTO 70 ELSE GOTO 100
70 PRINT "等根です X1=X2=";-B/(A+A):END
80 D=SQR(D)
90 X1=(-B+D)/(A+A): X2=(-B-D)/(A+A)
100 PRINT "X1=";X1,"X2=";X2:END
110 PRINT "根がありません":END
200 A=1:B=2:C=1:GOTO 40
300 A=1:B=2:C=5:GOTO 40
run
X1= 0 X2= 0
goto 200
等根です X1=X2= -1
goto 300
根がありません

```

図 2.1 ex21equation2.tx

```

D:\borland\#PBasic\#ex22equation3.txt
list
10 REM Cardanoの公式を応用して
20 REM 三次方程式 X^3+AX^2+BX+C=0 を解く
30 A=-6: B=11: C=-6: REM 例題データです
40 P=A*A/9-B/3
50 Q=-2*A*A*A/27+A*B/3-C
60 D2=Q*Q-4*P*P*P: D=SQR(ABS(D2))
70 IF D2>0 THEN GOTO 170
80 IF D2=0 THEN GOTO 150
90 ANG3=ATN2(D, Q):ANG=ANG3/3
100 CC=COS(ANG):SS=SIN(ANG):R=SQR(P)
110 PRINT "三つの実根があります"
120 X1=R*(CC+CC):PRINT "X1=";X1-A/3
130 X2=R*(-CC-SQR(3)*SS):PRINT "X2=";X2-A/3
140 X3=R*(CC-SQR(3)*SS):PRINT "X3=";X3-A/3:END
150 X1=(4*Q)^(1/3)
160 PRINT "等根です X1=X2=X3=";X1-A/3:END
170 U3=(Q+D)/2:U1=(ABS(U3))^(1/3)
180 IF U3<0 THEN U1=-U1
190 V1=P/U1:X1=U1+V1-A/3
200 PRINT "実数解は一つです X1=";X1:END
210 A=-5:B=8:C=-4:GOTO 40
220 A=-3:B=3:C=-1:GOTO 40
run
三つの実根があります
X1= 3
X2= 1
X3= 2

```

図 2.2 ex22equation3.txt

```

D:\borland\#PBasic\#ex23Newton.txt
list
10 REM Newtonの逐次近似法を使って
20 REM 三次方程式 X^3+AX^2+BX+C=0 を解く
30 A=-6:B=11:C=-6: XN=0: REM 例題です
40 ERR=10^(-6): REM 精度設定です
50 X=XN: 初期値です
60 Y=((X+A)*X+B)*X+C: DY=(3*X+2*A)*X+B
70 XN=X-Y/DY
80 PRINT "X=";X
90 IF ABS(XN-X)<ERR THEN END
100 X=XN: GOTO 60
110 XN=10: GOTO 50: REM 別の初期値です
run
X= 0
X= 0.54545454545454545
X= 0.848953210622453
X= 0.974674071023302
X= 0.999091548056949
X= 0.999998764691055
X= 0.999999999997711
goto 110
X= 10
X= 7.36125654450262
X= 5.61610701395443
X= 4.47379897271395
X= 3.74420450714993
X= 3.30588643643379
X= 3.08210386279587
X= 3.00848840787698
X= 3.00010598093566
X= 3.00000001684377

```

図 2.3 ex23Newton.txt

```

D:\borland\#PBasic\#ex24lineareq2.txt
list
10 REM 2直線の交点を求める問題は
20 REM 二元一次連立方程式を解きます
30 REM A*X+B*Y=P
40 REM C*X+D*Y=Q
50 REM パラメータはREAD文で読みます。
60 READ A,B,P,C,D,Q
70 DET=A*D-C*B
80 IF ABS(DET)=(10^(-6)) THEN GOTO 150
90 X=(P*D-Q*B)/DET
100 Y=(Q*A-P*C)/DET
110 PRINT "X=";X
120 PRINT "Y=";Y: END
130 PRINT "2直線は平行です": END
140 DATA 1,2,5, 3,-1,1
150 PRINT "2直線は平行です":END
run
X= 1
Y= 2

```

図 2.4 ex24lineareq2.txt

C3. グラフィックス

C3.0 一般

Plain_Basic は、グラフィックス表示専用の子ウインドウ（タイトルバーの名称は Canvas）を使います。親ウインドウの中と言う制限がありますが、位置と寸法を自由に設定できます。したがって、作図は、自動的に Canvas の寸法に合わせるように線図で描きます。この条件を考えて、Canvas 上のウインドウ座標系の設定と図形寸法の変換を次のようにしてあります。

- (1) カメラを使って、二次元の平面世界座標系（右手系）を視野に納めることを想定します。
- (2) カメラの光軸が狙う世界座標を(WX,WY)、視野に納める全横幅を WW とします。
- (3) カメラの設定をコマンド DPWIND WX,WY,WW で行います。デフォルトは、DPWIND 0,0,640 です。
- (4) 撮影は、平面世界座標に描く図形を擬似的なフィルムに記録すると考えます。
- (5) フィルムの横×縦の寸法比は 4:3 です。この比は標準グラフィックスモニタと同じです。
- (6) このフィルムを、Canvas に中心に最大に接するように焼きつけると考えます。
- (7) Canvas の横縦寸法比は、フィルムのそれと違いますので、天地または左右に余りが出ます。

作図は、線図で描き、塗り潰しの仕様を使いません。最も基本となるコマンドが DPMOVE/DPDRAW の対です。これを応用すれば殆どの作図ができます。ただし、基本的な図形のうち、円を描かせる DPCIRC を準備しました。線の長さは Canvas の寸法変化に対応して変化します。しかし、線の太さや線種（実線、破線など）、文字寸法などはビット単位の仕様で決められていますので、Canvas の寸法が変化しても変化しません。Canvas 上の文字寸法は、コマンドではなく、FONT メニューで変更します。

C3.1 多角形の作図

任意の正多角形を作り、そのすべての頂点間を線で結んだ単純な図形です。例題プログラムは、最初九角形を描きます。ラベル番号 180 で十六角形の設定に変えてラベル番号 30 から実行させたときの図形が右側の Canvas に作図されています。このプログラムで注目する処理は以下の項目です。

- (1) 角度の指定は度を使っています。
- (2) 以前に何かの作業をして OPTION RADIAN を呼んでいれば、違った図が得られます。
- (3) 配列の宣言を DIM 文(ラベル 50)で行い、その配列を ERASE 文(ラベル 160)で破棄しています。
- (4) Plain_Basic では、配列の成分表示に角括弧[]を使用します。
- (5) 作図のところで、FOR-NEXT 文を入れ子に使っています（図 3.1 ex31polygon.txt）。

C3.2 円の作図で描く紋様

円の中心座標と半径を種々変えて作図すると、幾何学的な紋様が得られます。例題では、まず一つの基準円を描きます。その円周上に中心を載せた幾つかの寸法の円を描かせます。その直径は、基準円周位置の x 座標です。円の内部を塗り潰す仕様を含めてありませんが、もし全体の外形図形だけを見たとき、どのような幾何学的方法で描いたかは簡単には分からないと思います。

（図 3.2 ex32circlear.txt）。

C3.3 乱数を応用した矩形の作図

矩形は、円と並んで、基本図形として頻繁に作図されます。多くのグラフィックスルーチンは、標準命令として矩形作図が準備されています。その場合、外形だけでなく、内部の塗り潰し仕様があるのが普通です。Plain_Basic では矩形描画のコマンドを含ませていません。矩形は四つの線分で構成しますので、これを内部のサブルーチンに組み上げて、それを GOSUB/RETURN 文で引用する例題を示します。矩形の対角線位置の座標を乱数関数 RND で発生させますが、矩形が Canvas の外にはみ出さないように、左右幅 600、上下高さ 600 で制限しました。例題の Canvas は、ほぼ正方形ですので作図の欠けは起こりませんが、横幅を広く取った Canvas 画面では天地が欠けます。（図 3.3 ex33rectangle.txt）。

C3.4 文字の作図例

文字列は、先頭文字の左下を指定した座標に合わせて作図します。文字は、システムで利用できるフォントであって、英字・数字・記号、及び漢字を描くことができます。文字寸法の変更は実行モードではなく、直接モードのときに、FONT-GRAPHICS メニューで行います。デフォルトは 9 ポイントです。

（図 3.4 ex34characters.txt）

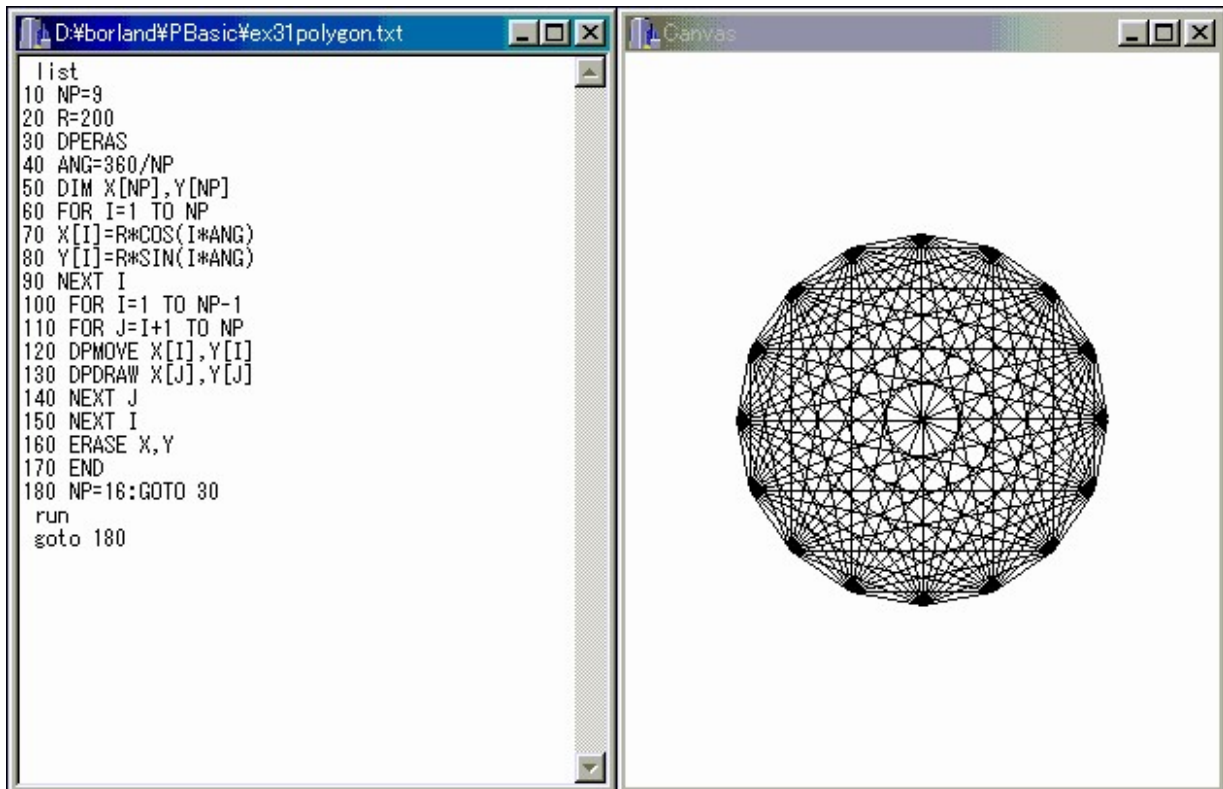


图 3.1 ex31polygon.txt

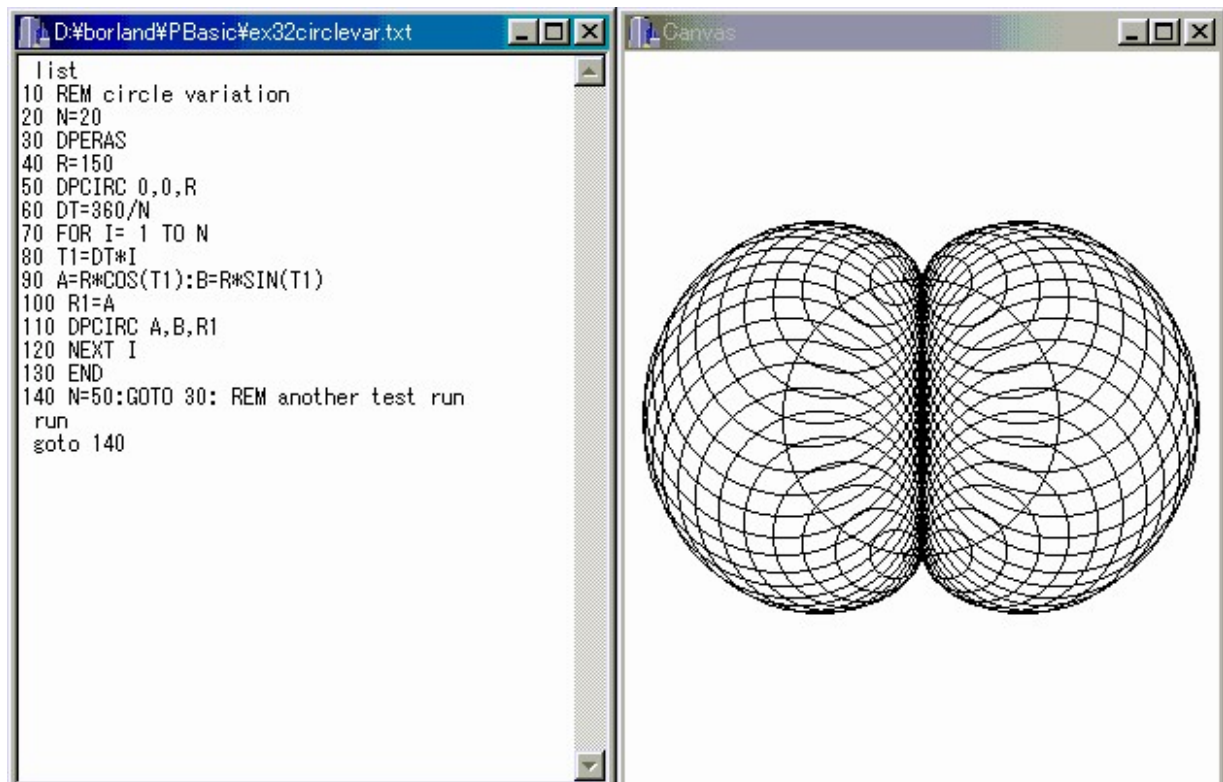


图 3.2 ex32circlevar.txt

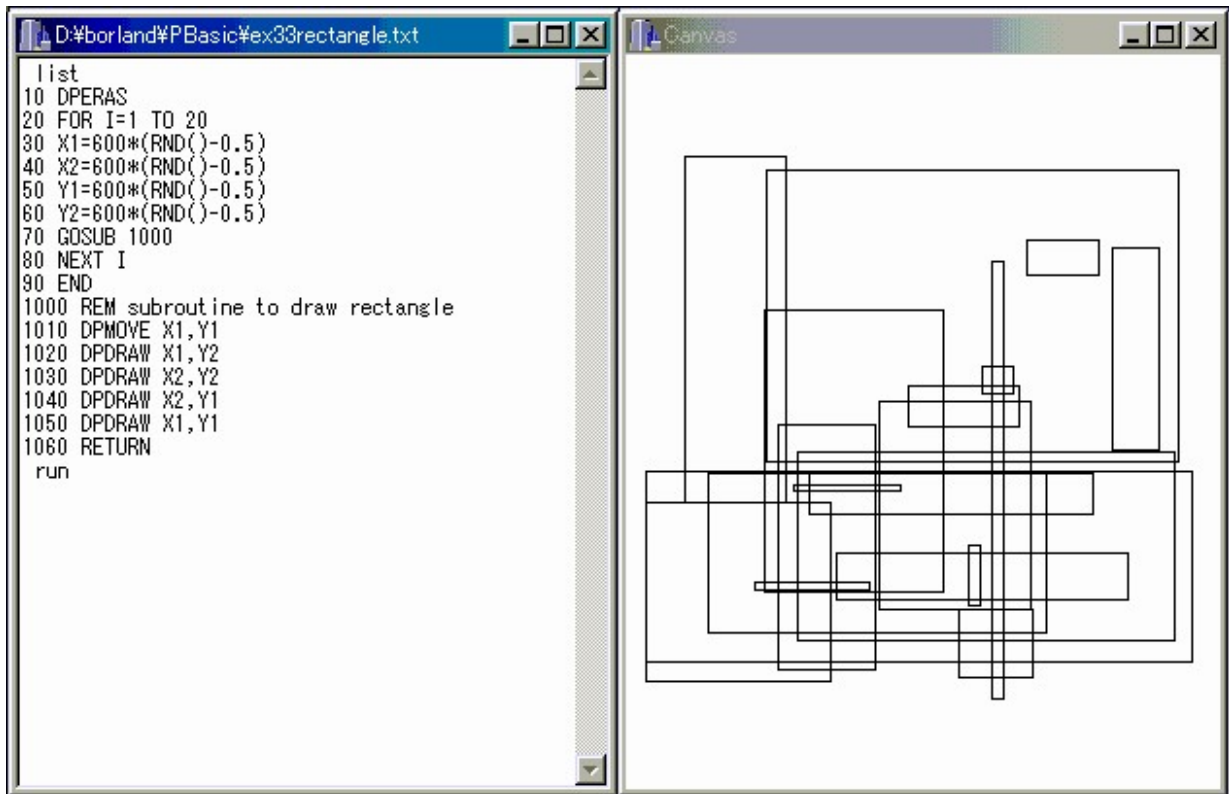


図 3.3 ex33rectangle.txt

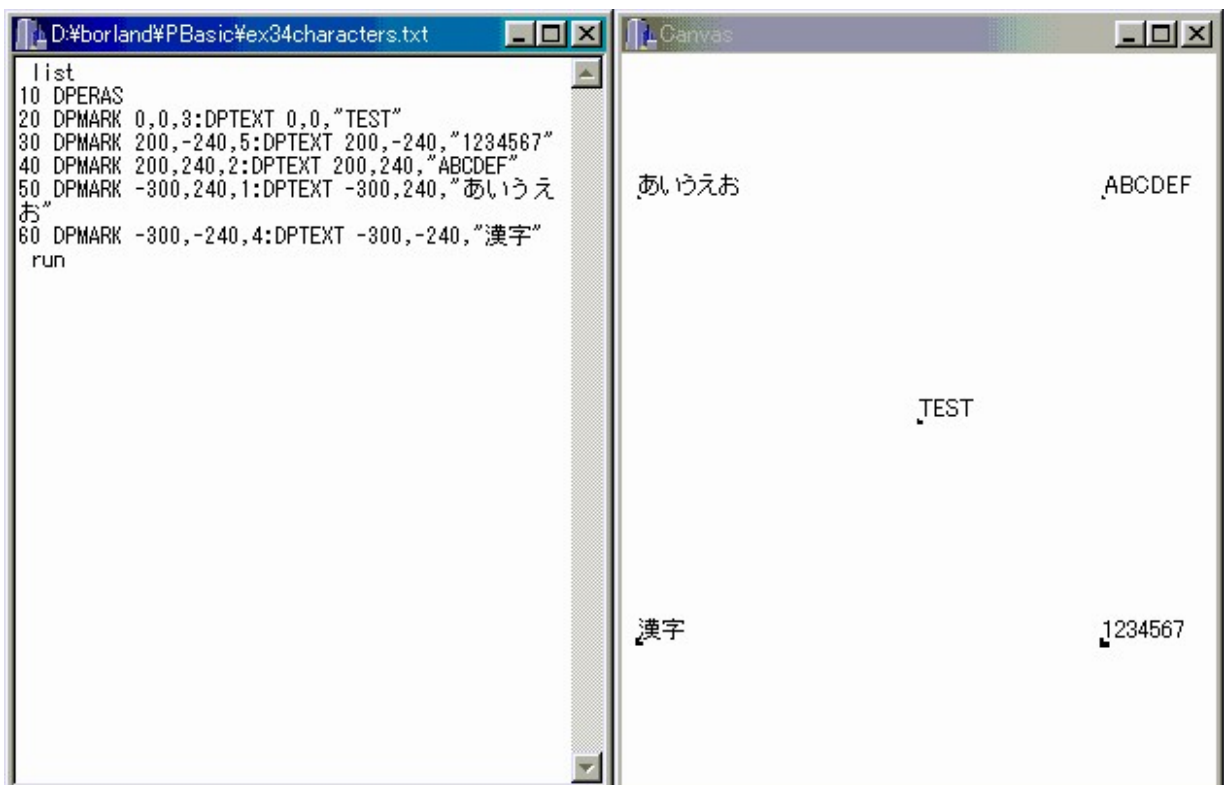


図 3.4 ex34characters.txt

C4. 関数の作図

C4.0 一般

二次や三次の方程式のグラフは曲線図形です。円もそうです。一般に曲線を描くときは何かの関数 $y=f(x)$ を考え、 x の値を変えながら複数の座標点を直線で結ぶような多角形で作図します。このとき、関数図形の納まる座標範囲 ($x_{min} < x < x_{max}$), ($y_{min} < y < y_{max}$) を考えた矩形範囲を決めます。これを矩形の枠で囲むのもよいでしょう。この枠が納まるように、DPWIND でウインドウを定義します。座標軸と尺度目盛が必要であれば描かせますが、尺度は DPTEXT を使って文字で記入します。ただし、例題では、座標範囲の決め方の過程を省いてあります。

C4.1 三次方程式($Y=X^3-3*X-4$)のグラフ

この式は、 $X=-2$ で $Y=-6$ 、 $X=3$ で $Y=14$ ですので、この範囲をグラフにすると横 5×縦 20 の縦長になります。見てくれをよくするため、横軸を 5 倍に拡大して作図することにします。そうすると、ウインドウの設定を「DPWIND 3.5, 10, 25」とし、 $(-2 < X < 3)$ に代えて、 $X=Z/5$ の関係に直し ($-10 < Z < 15$) を使ってグラフを描かせます。ウインドウの設定はデフォルト値と変わりますので、続けて別の作図をするときには注意が必要です。NEW コマンドを実行すれば、デフォルト値に戻ります(図 4.1 ex41draweq3.txt)。

C4.2 楕円の作図

円・楕円・放物線・双曲線は、代数的には二次曲線に分類されています。円や楕円は、sine, cosine などの三角関数を使わなくても、簡単な二次式を応用して作図できるのがコンピュータグラフィックスの面白さの一つです。楕円に外接する平行四辺形を考えます。一つの頂点とその両側で楕円と接する二点を取り出し、その間の 4 分の 1 楕円部分を描く曲線を繋ぎます。作図データは、三角形の 3 頂点 P1, P2, P3 の座標です。P1 で辺に接するように曲線が始まり、P3 で反対側の辺に接するように、途中の座標を計算して多角形で描きます。途中の座標の分割を細かくすれば滑らかな楕円 (1/4 部分) になります。パラメータ A を変えて GOTO 210 を実行すると残り楕円 (3/4 部分) を描きます。このプログラムは、事項の drawcurve.txt の中でサブルーチン化して、GOSUB で参照するようにしてあります。

(図 4.2 ex42ellipse.txt)

C4.3 イラストの作図

図形の基本は、自由な曲線を描くことに始まります。コンピュータグラフィックスでは、曲線は何かの数学関数を応用して座標を決め、短い線分を繋いで見た目で滑らかに見えるように作図します。実用的には、なるべく単純な数式で表される曲線を繋ぎます。その方法の一つは、最初に多角形で骨格を考え、その角を丸めるように二次曲線を当てはめます。上の C4.2 項で例示した楕円の作図がその例です。そこでは、3 角形の座標点で多角形の骨格を設計します。この処理を拡張して、NP 角形の骨格を使います。また行きっぱなしの場合と、最後の点と最初の点とを結んで、閉じた曲線ループを考えることにします。

まず、骨格となる多角形を設計します。これには、方眼紙を使って線図形を描き、座標を読み取ってデータ文を準備します。図 4.3 は、ex43panda.txt に整理したデータから多角形を作図させたものです。プログラムは drawpolyline.txt です。これら二つを MERGE して実行させました。リストは、図形の後に付してあります。なお、線種は点線の指定であって、直接モードで dpentx 3 にしました。なお、ウインドウの指定はプログラム中で行っていますが、他のデータを試すときには、画像がキャンバスに納まるように設定換えが必要です。

図 4.4 は、プログラム drawcurve.txt を使い、同じく ex43panda.txt と MERGE して実行させたものです。線の太さは、直接モード状態で、dpensz を呼んで 2 ピクセル幅にしてあります。

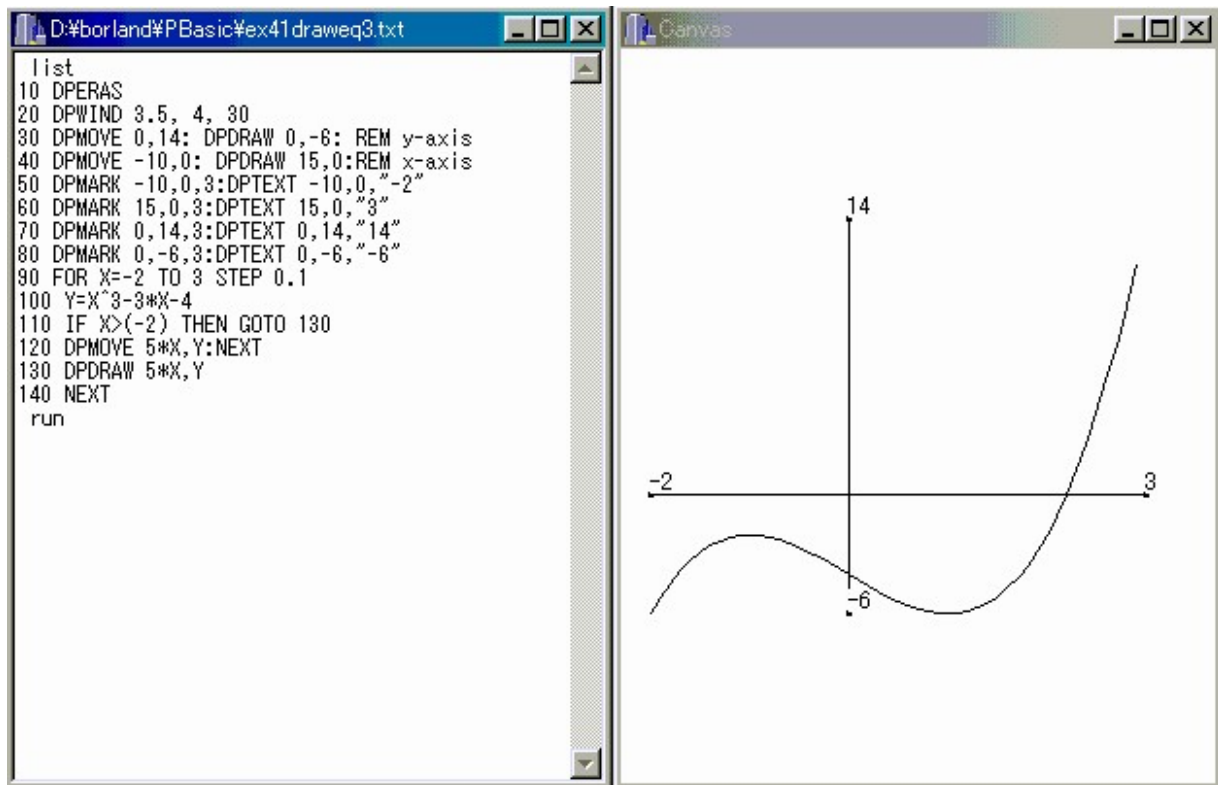


图 4.1 ex41draweq3.txt

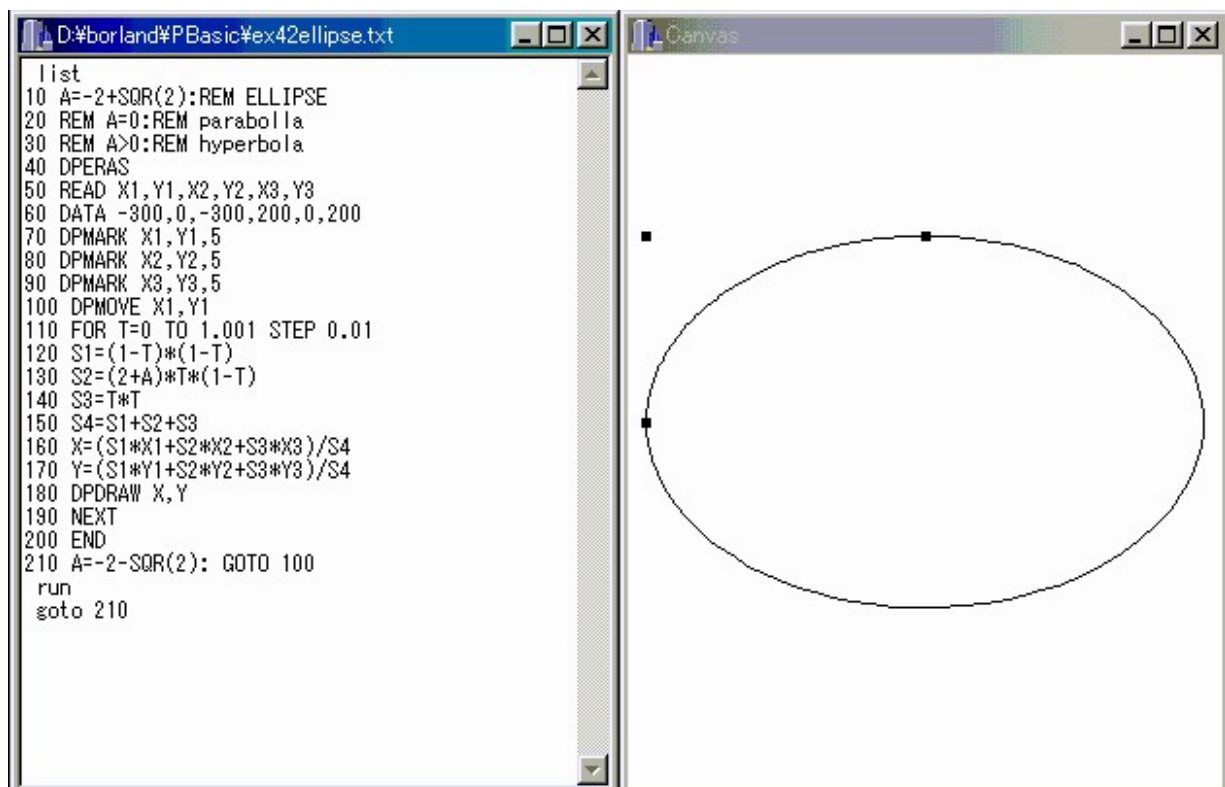
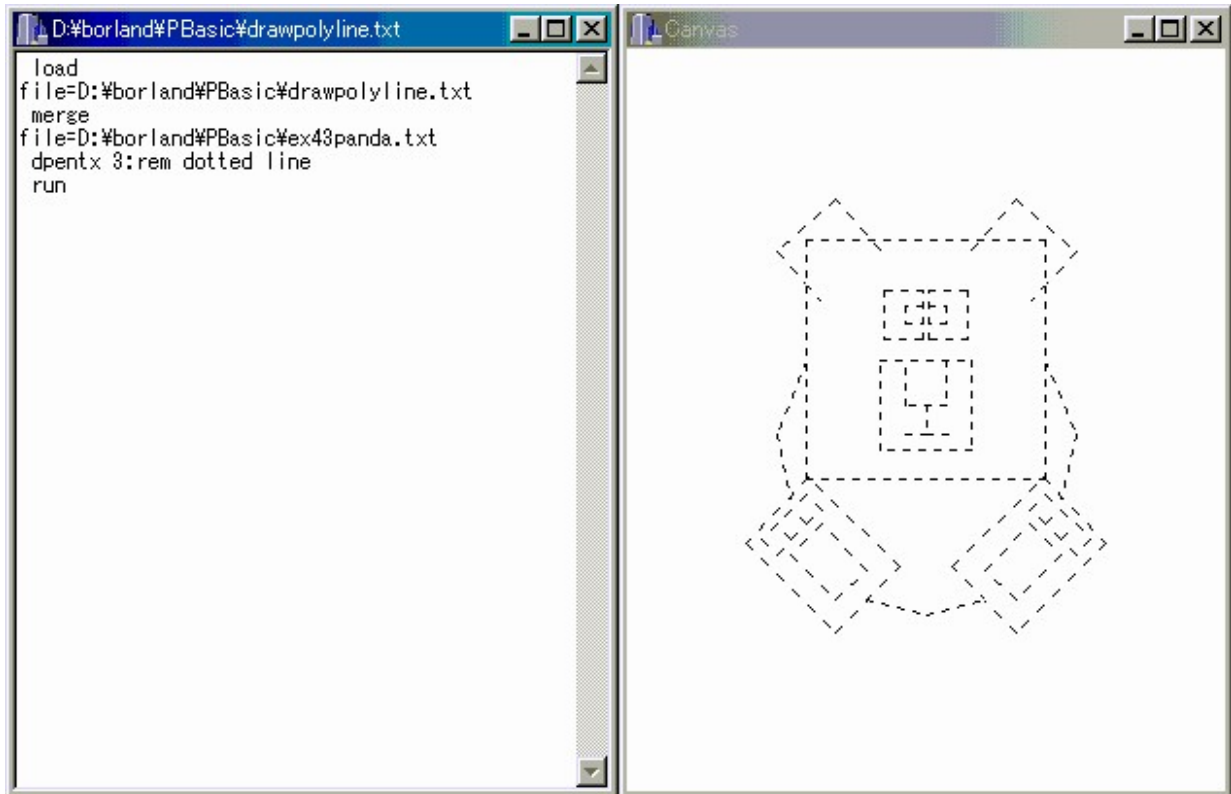
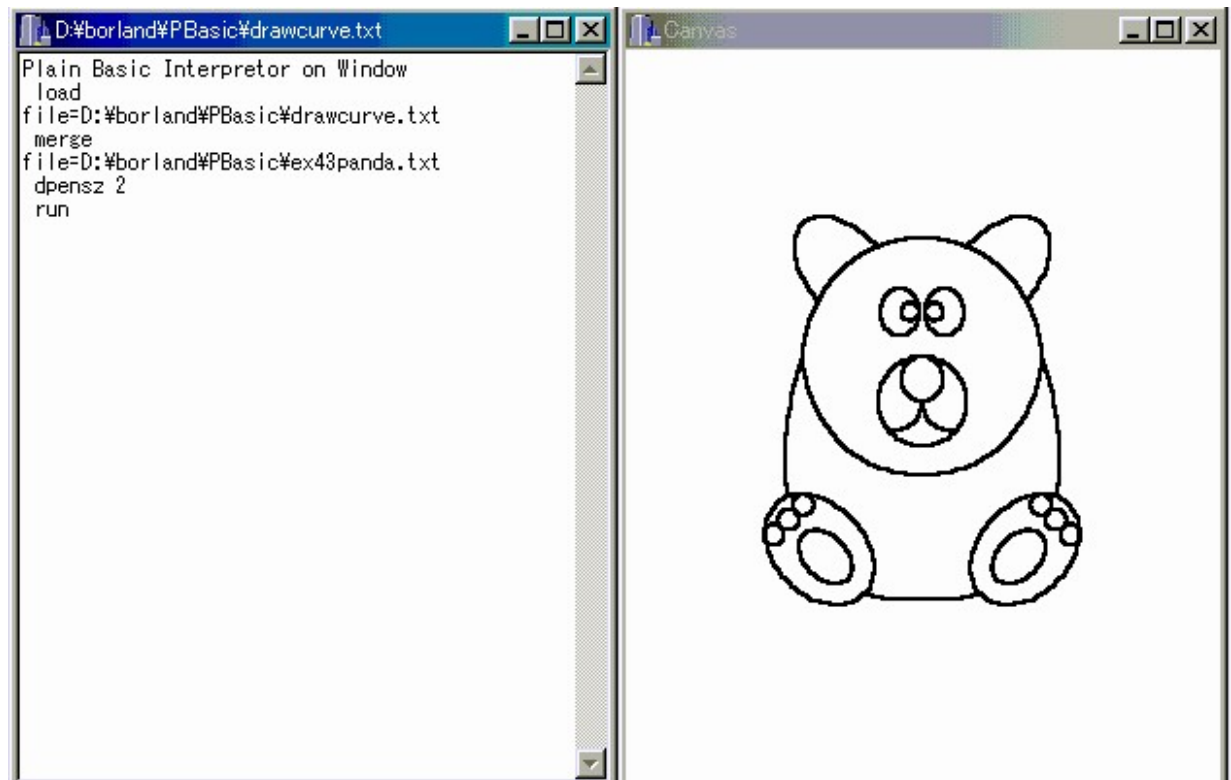


图 4.2 ex42ellipse.txt



4.3 drawpolyline.txt



4.4 ex42drawcurve.txt

表 C1 drawpolyline.txt のリスト

```

10 DIM P[2,10]: REM reserve 10 points
20 REM draw polyline
30 DPERAS: DPWIND 0,-20,200
40 READ P: REM ===== Main loop
50 IF P[1,1]=9999 THEN GOTO 9010
60 NP=1: IOPEN=1
70 REM: ///loop-A/// to find number of points NP
80 IF P[1,NP+1]=9000 THEN GOTO 1000
90 IF P[1,NP+1]=9900 THEN GOTO 1010
100 NP=NP+1: IF NP>10 THEN GOTO 9000
110 GOTO 70 :REM /// return to loop-A
120 REM -----
1000 IOPEN=0
1010 K=1
1020 DPMOVE P[1,1],P[2,1]
1030 DPDRAW P[1,K+1],P[2,K+1]
1040 K=K+1
1050 IF K<NP THEN GOTO 1030
1060 IF IOPEN=0 THEN GOTO 40
1070 DPDRAW P[1,1],P[2,1]: GOTO 40
9000 PRINT "data error"
9010 END

```

表 C2 ex43panda.txt のリスト

```

5000 DATA 15,37,30,54,50,36,35,19,9000/
5010 DATA -15,37,-30,54,-50,36,-35,19,9000/
5020 DATA 1,23,14,23,14,7,1,7,9900/
5030 DATA -1,23,-14,23,-14,7,-1,7,9900/
5040 DATA 1,18,7,18,7,12,1,12,9900/
5050 DATA -1,18,-7,18,-7,12,-1,12,9900/
5060 DATA 0,-15,0,-25,10,-25,9000/
5070 DATA 0,-15,0,-25,-10,-25,9000/
5080 DATA 40,0,50,-25,45,-47,9000/
5090 DATA -40,0,-50,-25,-45,-47,9000/
5100 DATA 39,-39,60,-61,30,-91,9,-69,9900/
5110 DATA -39,-39,-60,-61,-30,-91,-9,-69,9900/
5120 DATA 40,-45,45,-50,40,-55,35,-50,9900/
5130 DATA -40,-45,-45,-50,-40,-55,-35,-50,9900/
5140 DATA 45,-50,50,-55,45,-60,40,-55,9900/
5150 DATA -45,-50,-50,-55,-45,-60,-40,-55,9900/
5160 DATA 50,-55,55,-60,50,-65,45,-60,9900/
5170 DATA -50,-55,-55,-60,-50,-65,-45,-60,9900/
5180 DATA 35,-55,45,-65,30,-80,20,-70,9900/
5190 DATA -35,-55,-45,-65,-30,-80,-20,-70,9900/
5200 DATA -40,40,40,40,40,-40,-40,-40,9900/
5210 DATA -7,0,7,0,7,-15,-7,-15,9900/
5220 DATA -15,0,15,0,15,-30,-15,-30,9900/
5230 DATA -20,-80,0,-85,20,-80,9000/
5240 DATA 9999/
5250 DATA -1,23,-14,23,-14,7,-1,7,9900/
5260 DATA 1,18,7,18,7,12,1,12,9900/
5270 DATA -1,18,-7,18,-7,12,-1,12,9900/
5280 DATA 0,-15,0,-25,10,-25,9000/
5290 DATA 0,-15,0,-25,-10,-25,9000/
5300 DATA 40,0,50,-25,45,-47,9000/
5310 DATA -40,0,-50,-25,-45,-47,9000/
5320 DATA 39,-39,60,-61,30,-91,9,-69,9900/
5330 DATA -39,-39,-60,-61,-30,-91,-9,-69,9900/
5340 DATA 40,-45,45,-50,40,-55,35,-50,9900/
5350 DATA -40,-45,-45,-50,-40,-55,-35,-50,9900/
5360 DATA 45,-50,50,-55,45,-60,40,-55,9900/
5370 DATA -45,-50,-50,-55,-45,-60,-40,-55,9900/
5380 DATA 50,-55,55,-60,50,-65,45,-60,9900/
5390 DATA -50,-55,-55,-60,-50,-65,-45,-60,9900/
5400 DATA 35,-55,45,-65,30,-80,20,-70,9900/
5410 DATA -35,-55,-45,-65,-30,-80,-20,-70,9900/
5420 DATA -40,40,40,40,40,-40,-40,-40,9900/
5430 DATA -7,0,7,0,7,-15,-7,-15,9900/
5440 DATA -15,0,15,0,15,-30,-15,-30,9900/
5450 DATA -20,-80,0,-85,20,-80,9000/
5460 DATA 9999/

```

表 C3 drawcurve.txt のリスト

```

10 DIM P[2,10]: REM reserve 10 points
20 A=-2+SQR(2):REM draw with ellipse
30 DPERAS: DPWIND 0,-20,200
40 READ P: REM ===== Main loop
50 IF P[1,1]=9999 THEN GOTO 9010
60 NP=1
70 REM: ///loop-A/// to find number of points NP
90 IF P[1,NP+1]=9000 THEN GOTO 1000
110 IF P[1,NP+1]=9900 THEN GOTO 2000
120 NP=NP+1: IF NP>10 THEN GOTO 9000
130 GOTO 70 :REM /// return to loop-A
140 REM -----
1000 K=1: REM /// open curve drawing
1010 IF NP<2 THEN GOTO 9000
1020 X1=(P[1,K]+P[1,K+1])/2: Y1=(P[2,K]+P[2,K+1])/2
1030 IF K<>1 THEN GOTO 1050
1040 X1=P[1,K]: Y1=P[2,K]
1050 X2=P[1,K+1]: Y2=P[2,K+1]
1060 IF NP=2 THEN GOTO 1130
1070 X3=(P[1,K+1]+P[1,K+2])/2: Y3=(P[2,K+1]+P[2,K+2])/2
1080 IF (K+2)<>NP THEN GOTO 1100
1090 X3=P[1,NP]: Y3=P[2,NP]
1100 GOSUB 3000
1110 IF (K+2)=NP THEN GOTO 40: REM to Main
1120 X1=X3: Y1=Y3: K=K+1: GOTO 1050
1130 DOMOVE X1,Y1:DPDRAW X2,Y2: GOTO 40: REM to Main
1140 REM -----
2000 K=1: REM /// closed curve drawing
2010 IF NP<=2 THEN GOTO 9000
2020 X1=(P[1,1]+P[1,NP])/2: Y1=(P[2,1]+P[2,NP])/2
2030 X0=X1:Y0=Y1
2040 X2=P[1,K]: Y2=P[2,K]
2050 X3=(P[1,K]+P[1,K+1])/2: Y3=(P[2,K]+P[2,K+1])/2
2060 IF K<>NP THEN GOTO 2080
2070 X3=X0: Y3=Y0
2080 GOSUB 3000
2090 IF K=NP THEN GOTO 40 :REM to Main
2100 X1=X3: Y1=Y3: K=K+1: GOTO 2040
3000 REM -----
3010 REM subroutine curve drawing
3020 REM parameters A, X1, Y1, X2, Y2, X3, Y3
3030 REM shall be set in the Caller
3040 REM A=-2+SQR(2):REM ellipse
3050 REM A=0 :REM parabolla
3060 REM A>0 :REM hyperbola
3070 DPMOVE X1, Y1
3080 FOR T=0 TO 1.001 STEP 0.05
3090 S1=(1-T)*(1-T)
3100 S2=(2+A)*T*(1-T)
3110 S3=T*T
3120 S4=S1+S2+S3
3130 X=(S1*X1+S2*X2+S3*X3)/S4
3140 Y=(S1*Y1+S2*Y2+S3*Y3)/S4
3150 DPMOVE X, Y
3160 NEXT
3170 RETURN
9000 PRINT "data error"
9010 END

```