1.1 文字の画面表示　　ver. 0.3

プログラム言語や計量ソフトを新しく学ぶ際に、まず最初の乗り越えなければならない関門は、まず実際に一人で何か小さなプログラムを動かしてみることでしょう。これは、GAUSS に限ったことではありませんが、何か動作をする最小単位のプログラムを知ることは、最も新鮮で、感激をもって目にできるはずです。その手始めに、一番簡単な文字を画面にプリントアウトすることから始めましょう。（当講座は Windows 上のものです。）

GAUSS をクリックして起動後、まず EDIT の中で実際に今から作るプログラムの名前をつけましょう。その際に、プログラムの保存場所は、各自が持ち寄ったフロッピーDISK または、それに類似する媒体が適当でしょう。例えば、D ドライブに sample1.txt というプログラムファイルを作るのなら、EDIT の中でファイルの場所を(D:)にかえてから、sample1.txt という名前をつけます。注意したいのが、必ず.txt という拡張子をファイル名につけることを忘れないようにしてほしいということです。拡張子をつけなければ、プログラムファイルは、GAUSS 内部からしか開かない未定義ファイルになってしまいます。無論、アスキー形式や各種ワープロ形式の拡張子をつけて適宜保存することも可能です。なにもワープロソフトが入っていないコンピューターでも後で自由に外から開くことができる点で.txt 形式をお勧めします。

ポイント　保存ファイルには拡張子（.txt）などを必ずつける。

では、ワークスペースとなるプログラムファイルの名前が決まったところで、次のようなプログラムを書いてみましょう。例えば、「Howdy, folks!」と画面上にプリントするプログラムを書きましょう。

プログラム

```
print “Howdy, folks.”;
```

画面結果

```
Howdy, folks!
```

このプログラムは、GAUSS 上の RUN コマンドを押すことによって、実行されて、画面上に”　“内の文字がプリントされることになります。（正確には、”　“内の文字が画面上にプリントされた上で改行されるという言い方が正しい。Ｃ言語などのプログラムでは改行もプリントの命令内に指定してやらないと何もしないが、GAUSS では改行という作業が print 命令のデフォルトになっている。）少し一般常識からすれば奇異に感じるかもしれませんが、たいていのプログラム言語では、print とは特にオプションをつけない限り、画面上に表示するという命令を指します。以下では、print の表示と印刷という日本語を特にこ

とわりのないかぎり、「画面に表示する」という意味として読者は読んでもらいたい。最後に、１連の命令の後には必ず；というマークをつけることに注意してほしい。；のマークを１つでも忘れると、これから作成していくプログラムはエラーを起こして動かないので細心の注意をもってことにあたりましょう。（カードで読み取りをしていた時代からあるFortran などの古い言語では、プログラムは１行一命令が原則であるが、Ｃ言語をはじめとする最近の言語では、プログラムに行の概念がありません。同様に GAUSS でもプログラムに行の概念はありません。したがって、１つの命令を書くと必ずその後には；のマークを忘れずに書く必要があります。）

ポイント　print の１つの機能には” “内の文字を画面に表示することである。
ポイント　各命令の最後には必ず；のマークをつけることを忘れない。

　では、次に２行にわたって文字を画面表示する応用をしてみましょう。これをすることにより、print には改行が含まれているということがわかるでしょう。「Howdy, folks!」と「How is it going?」という２つの文章を画面表示してください。

プログラム

```
print “Howdy,folks.”;
print “How is it going? ”;
```

画面結果

```
Howdy, folks!
How is it going?
```

うまく表示されたでしょうか？”　“内に含まれる文章は半角の英語であるかぎり、ピリオドが付こうと、クエスチョンマークが付こうと、関係ありません。実際に何か違う文面の英語を入れて遊んでみてください。
　では、２つの英文を連続して１行に書けないでしょうか？答えは下のようになります。

プログラム

```
print “Howdy,folks.”;;
print “How is it going? ”;
```

画面表示

```
Howdy, folk. How is it going?
```

ポイント　print 命令の後の；；は２つ組で改行するデフォルトを無効にする機能がある。

表示画面がなんだか混み合ってきました。前回までの結果をすべて消して、今回の結果だけを表示するプログラムにしてみましょう。

プログラム

```
cls;
print “Howdy,folks.”;;
print “How is it going? ”;
```

ポイント　cls はクリヤースクリーンの略で、それまでの画面表示を消去する命令。

または、GAUSS には行の概念がないのですから、次のようにもできます。

```
cls;
print “Howdy,folks.”;; print “How is it going? ”;
```

なお、” “内のスペースは表示結果に影響を与えますが、print と；のあいだのスペースや、命令と命令のスペースは何も結果に影響を与えません。見る人の立場（あるいは論文付属のプログラムであれば、レフリーの立場）を十分考えて、見やすく書きましょう。

最後に、いままでやってきたことをプログラムにまとめて、レポート形式にして保存しましょう。その前に、もう１つ。見る人に読みやすいプログラムを書くには、このプログラムが何をするプログラムなのかをプログラムの冒頭に注釈の形で書くこと、それから、プログラムの各行が何を示しているのかを注釈の形で書くことの２つが必要になります。それには、GAUSS は、Ｃ言語などと同じように、/*　　*/の内に注釈を書くという決まりになっています。もちろん、この方法が一般的なのですが、その他に、@　　@の内に注釈を書いてもかまいません。適宜、使い分けましょう。１つのプログラムを見やすくする方法に、文頭のプログラム名や作者名を書くとき、あるいはプログラムのブロックが何をしているのかを表すには、伝統的な/*　　*/を使い、それ以外のそれぞれの命令が何を示すかには、@　　@をその後ろにつけておくのもよく行なわれる方法です。

プログラム

```
/*---------------------------------------------- */
/* program 1.1          by Taro Bush */
/*---------------------------------------------- */
cls;                          @   Clear the screen.                    @
print “Howdy,folks.”;       @   Print a string to screen and return.   @
print “How is it going? ”;  @   Same as above                          @
/*              without return              */
```

```
print “Howdy,folks.”;;        @   Print a string to screen without return.   @
print “How is it going? ”;   @   Print a string to screen and return.        @
```

このようなプログラムを清書した上で、sample1.txt という名前をつけて、指定されている適当なドライブまたはディレクトリーに保存してみてください。繰り返しますが、ドット以下の拡張子なしでもファイルは保存できますが、GAUSS の内部からしか、そのファイルはのぞけなくなってしまいます。もちろん、.g や.asc などという風に拡張子を付けることも慣用化されてはいますが、やはり GAUSS 内部からしか見られない、WINDOWS 上ではただの無定義ファイルになってしまいます。保存したファイルをノートパッドなどで開くと、フォントの大きさの関係やタブやスペースの使い具合から多少行はずれてしまいます。しかし、これをまた修正すると、今度は GAUSS の中から開いたときにずれてしまいますので、そのままにしておいた方が賢明でしょう。

ここで、意外に知られていないコメントのつけ方の落とし穴を紹介したいと思います。次のプログラムを見てください。

プログラム（動作不可例）

```
@@-----------------------------------------------@@
@   program 1.1          by Taro Bush   @
@@-----------------------------------------------@@
cls;                              @   Clear the screen.                          @
print “Howdy,folks.”;          @   Print a string to screen and return.       @
print “How is it going? ”;   @   Same as above                                 @
/*              without return              */
print “Howdy,folks.”;;        @   Print a string to screen without return.   @
print “How is it going? ”;   @   Print a string to screen and return.        @
```

このプログラムは動きません。違うところは、題目のところに /*　*/ではなくて@　@を使っただけです。GAUSS の文法的にも、/*　*/を幾十にもすることは許されています。@@が重なってはいますが、@の数も偶数個あって、理論的にはコメントが成立するはずです。しかし、このプログラムは、最初の命令である cls から動きません。このように、@　@の注釈が便利であるからといって、@のマークで注釈自体を複数個つなげてしまったり、無数の@マークで囲んでしまうと、全然うんともすんともいわないプログラムになってしまいます。@を使うときには、注釈の最初と最後が分かりにくいこともある点からも、命令文の直後の注釈の使用にとどめ、絶対に重ねたり無数のマークで注釈を囲んではいけません。

ポイント　注釈は/*　*/の中に書くようにする。
ポイント　各行の注釈は@　@の中に書いてもよい。ただし@のマークを重ねない。

なお、実際のプログラム、特に、後で共用することを目的に作られた procedure には、その冒頭に長大なプログラムの使用法や copyright、それからどのラインにプログラムのどの部分があるのかという目次がついていることが多い。例えば、GAUSS 標準の ols.src という内部プログラムは、次のように冒頭が注釈文で書かれています。

```
/*
** ols.src - Least Squares Regression
** (C) Copyright 1988-1998 by Aptech Systems, Inc.
** All Rights Reserved.
**



**    Globals:       __altnam, __output, __row, __miss __con, _olsres, _olsrnam,
**                    indices2(), maxvec(), indexcat(), dotfeq()
**
**    See Also:     olsqr
*/
```

上のように、/* ではじまって、*/で終わるのですが、途中は見やすいように**のマークが各行のはじめに置かれています。注釈が長くなれば、**のマークを便宜上、注釈がまだ続いているということを示すために使うことも慣例化されています。最後にこれらのプログラムは、すべて半角の英語で書いてください。全角は英語のソフトでは、認識不可能です。注釈も英語で書くことが、国際間の学問の発展継承に重要です。

[注]

print 文なしでも、” abcdefg “;と文字を引用符にくるむことによって print 文の命令と同様のことはできますが、後述の文字列の扱いと混同されて、たいへん非常に読みにくいプログラムになります。新しいバージョンの GAUSS のサンプルプログラムにもこの方法はとり入れられていますが、昔からある公式の正当なコーディング方法ではありません。また、同様に、変数の中味を、たとえば変数名 a=1;というように a に１が入っているならば、a; などとして直接 print 文なしでその内容を表示できます。これらの方法は、プログラムの構造を非常にわかりにくくする方法なことが１つと、後述するエラー処理などのように変数にメッセージを伴っているものをある点がもう１つ、などの理由でプログラム全体の見通しが悪くなるので慎むべきです。（現に、公式マニュアルにはそれらのやり方の例の記述はなく、単に”implicit”なやり方であるという記述に抑えています。）たまたま、GAUSS の文字

列を扱う構造と、変数を扱う構造が、そうなっているにすぎません。必ず、print 文を使うように心掛けたいものです。直接 GAUSS を正式に教わったという人よりも Journal で他のソフトとの速度比較などの記述で、print 文を使わないプログラムが掲載されていたために、print 文を用いないやり方が急速に広まったと思われます。GAUSS の中心的な作者の Ronald Schoenberg 教授や、GPE2 の作者の Lin 教授らクリエーター側が、print 文を用いて正式な方法でコーディングしていることを最後に付け加えておきたいと思います。行下げを行なう階層的なプログラム、procedure を用いたプログラム、print 文や errorlog 文を直接（フォーマット系の printfm,printfmt 文も含む）使ったプログラム、それらをすべて兼ね備えることによって、万人に読みうる方法になると言えます。GAUSS は決して「べた書き」をするような記号言語ではありません。近代言語は、print や display などをそのまま print～として使うグループと丸括弧で print( )などとするグループに分かれます。そのうちの前者のグループに属するのが GAUSS と言えます。

	正統的な記述法	亜流な記述法
コメント表示	print “This is a comment”;	“This is a comment”;
改行	print;	?; または “ “;
変数値の表示	a=1+2; print a;	a=1+2; a;

このうち、？のマークはほかの言語ではコメント行を表すしるしであって、誤解をまねくとともに、GAUSSX というGAUSS のアプリケーション言語ではコメント行を表す記号としてリザーブされた命令です。使用は厳に慎むべきです。また、a;などと記述する方法は制御文と混同され見通しの悪いプログラムになります。後に扱う procedure 内の retp 文と合わせて、print 文と retp 文はプログラムの流れの中でどこで外部に出力されたり抜けたりする目印を与える働きがあります。命令 print というところは必ずつけてプログラムするようにしてください。