



QB-MONITOR

QB-99'ERS U.G. NEWSLETTER

SEPTEMBER 1989

The QB MONITOR is the Newsletter of the QB-99'ers User Group, is printed Sept. thru June and sent in exchange for other User Group Newsletters. Send Exchange Newsletter to Frank Catty, Queensborough Community College, Bayside, NY 11364. Credit original sources.

The QB 99'ers meets the second Saturday of each month September through May, at Queensborough Community College, Bayside New York, room S225, at 2 P.M. Calendar at right shows dates

September 1989						
S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

October 1989						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

ATTENTION:

QB 99'ers Juniors Users Group now forming. See application form in this newsletter.

CONTENTS:

PAGE

- 1) QB JUNIORS GROUP
- 2) STYLE-A-LINE UPDATE
- 3-6) DECISIONS
- 7) CATALABEL REVISITED
- 8-10) TIPS FROM THE TIGERCUB

QB MONITOR ~ QB-99'er NEWSLETTER

TI FOR KIDS!

THE QB 99'ERS will be having special meetings starting in March to meet the special needs of children. These will be before the start of our regularly scheduled meetings.

These special meetings will be free to children of QB 99'ers members. Children under the age of 10 (ten) MUST be accompanied by an adult.

Please fill out the form below and bring it with you to:

Room S 225 (science building) at Queensborough Community College

-- 1 PM Saturday March 14 --if you are interested in attending..

the CHILDREN'S QB 99'ERS GROUP REGISTRATION FORM

MEMBERS NAME

CHILDREN NAMES

TELEPHONE #

AGES

Do you have TI Logo? (Y/N) Bring the manual with you if you have it!

If you are not a member and would like to join our group for your children, simply become a member by coming in with your child on March 14 at 1 PM room S 225 Queensborough Community College, Bayside, New York. There you will learn about our group and its requirements.

STYLE-A-LINE UPDATE

By Ed Machonis

If you are using STYLE-A-LINE you may have noted its annoying habit of sending a Line Feed each time a control code is sent to the printer. This can be eliminated by opening the printer file as PID.LF. Since we require a line feed each time we print a line of text, a CHR\$(10) must be sent at the end of the text. The changes are shown below for Lines 3 and 9 in a manner that maintains the program's Tiny Gram status.

```
3 OPEN #1:"PID.LF",VARIABLE
132 :: L$=CHR$(10)
```

```
9 IF TRY$="ZZZ" OR TRY$="zzz
" THEN 4 ELSE TEX$=TRY$&L$ :
: PRINT #1:TEX$ :: GOTO 8
```

DECISIONS

By Ed Machonis

Decisions, Decisions, Decisions! It seems like almost every day we are faced with a new batch of problems requiring decisions. Cheer up, help is at hand in the form of your trusty (rusty?) TI-99. "What", you say, "let a computer make my decisions? Has it come to this?"

Not to worry. No, the TI-99 won't make your decisions, but it sure will help you arrive at a decision you will feel comfortable with. The TI-99 will ask you for all the factors that bear on your decision, the relative importance of each, and what the alternative choices are. It will help you look at your problem from many angles and make you think more deeply about it than would probably be the case if you were tackling the problem alone. It will weigh the relevant factors and alternative courses of action in a manner that most of us wouldn't dream of attempting and which would take hours if we did. But think about it, isn't this why we bought the computer in the first place?

When I first got my TI-99, one of the tools I used to learn Basic, was to translate programs written for other computers. My computing equipment then consisted of the console, a cassette recorder and a TV. One of the books I found at that time was Basic Programs for the Pet Computer, which contained the program Decide, written by Phil Feldman and Tom Rugg back in the early days of 1978. I translated the program, added a few frills and used it to make some important career related decisions.

Over the years whenever my wife suggested listing the pros and cons of a matter under consideration, I'd boot up DECISIONS and it would generally point out the way to go. Recently, faced with a very important decision, the scenario was repeated. While I had the program out, I thought, "Why am I keeping this unique gem to myself, why not share it with others through the newsletter?"

When I examined the listing, (which took up 3 pages, 3 columns per page) I was more than a little surprised. My translation was in Console Basic and a finer dish of spaghetti was not to be had this side of Bologna. I thought, "My standards sure have changed in the 7 intervening years." I didn't want to run the listing and admit I had anything to do with it, even though it probably reflected the work of the original authors.

The best way of straightening the program out seemed to be to recast it in extended basic, which I proceeded to do. When I was nearly through, I wondered if I still had a copy of the original listing for the Pet. Sure enough I did. Examining it brought on surprise number two.

My extended basic version, except for syntax differences, very closely resembled the original program. All that spaghetti code I was so critical of had been introduced by me in translating multi-statement lines into Console Basic. The original authors had a lot of IF-THEN-ELSEs which I compounded by having to break them into individual statement lines. Live and Learn. I'm sure glad I eventually got Extended Basic.

But on to the program itself. As the original authors put it....."This program can help you make decisions involving the selection of one alternative from several choices. It works by prying relevant information from you and then organizing it in a meaningful, quantitative manner. Your best choice will be indicated and all of the possibilities given a relative rating."

The initial screen requires you to choose one of three categories that best fits your decision problem.

- 1) Choosing an Item or Thing.
- 2) Choosing a Course of Action.
- 3) Making a Yes or No decision.

If you choose an Item, you will be asked what type of item it is.

If your decision fits one of the first two categories, you will have to list all of the possibilities under consideration. While up to 10 possibilities may entered, you must enter at least two. If you have trouble making decisions with only one possibility, you need more help than this program can provide.

Next you will be asked for all the different factors that are important to you in making your choice. For instance, if you were thinking of purchasing a new car, Style, Price, Service, Reliability, Standard Equipment and Safety would probably be considered. The number of factors is limited to 10, however, you can run the program with just a single factor being considered, although the result will be fairly obvious.

DECISIONS.....Text Page 2

You must next decide which single factor is the most important to you. This factor is given a value of 10 and you must then rate the importance of each of the other factors with respect to the most important factor, using a scale of 0 to 10. a factor equally important would carry a value of 10, while one only half as important might be rated 5.

When you are done, the program will evaluate the relative weight of each importance factor with respect to the total weight of all factors.

Next you will be asked to rate each of the decision alternatives with respect to each of the importance factors. One of the alternatives will be assigned a value of 10 and you will be asked to rate each of the other alternatives with respect to each importance factor, depending on how much better or worse you think it is.

Let's say you were considering a new computer (Byte Your Tongue Ed) and the alternatives were a Geneve, an IBM Clone and a Turbo XT. Let us also say that one of the importance factors was Price. The program would assign a value of 10 to the Geneve (if it was the first one entered) and ask you to rate the IBM Clone pricewise with respect to the 10 assigned to the Geneve. Depending on the configuration of the Clone being considered you might rate it B if it cost more or 12 if it cost less, the values depending on how much more or less it cost. Then you would rate the Turbo XT in the same manner.

Next (if it was entered as a factor) Ease of Use would be rated, with 10 again being assigned to the Geneve and you rating the Clone and the Turbo XT on how they stacked up against the Geneve.

Undoubtedly Ease of Use carries a different weight of importance than does Price. You can see the mathematical complexities we are getting into, including two dimensional arrays. Not to worry, that's why we bought the computer. I don't think you would want to try this with pencil and paper, or even with a calculator. Your TI-99 will take all your inputs for each possible choice and importance factor and determine which choice is BEST FOR YOU.

It will list all the alternatives and their final relative rankings, assigning a normalized value of 100 to the best choice.

If you had chosen Making a Yes or No Decision, there of course would be no other decision alternatives involved, only Deciding Yes and Deciding No. Deciding Yes will be rated 10 for each importance factor and you will be asked to rate that factor with respect to Deciding NO; assigning it a value higher or lower than 10 depending on how much you think it is better or worse than Deciding Yes.

The original authors suggested some modifications.

1. Increase the number of possible Alternative and/or Factors by changing the array dimensions in Line 120. (Also change Lines 310, 320, 430 440 to suit.)

2. Allow the user to review his numerical input and modify it if desired.

3. Insights into a decision can often be gained by a sensitivity analysis. Modify the program to allow a change of input after the regular output is produced. Then recalculate the results based on the new values.

To this I would add....Modify the program to provide a print out of the final ranking of the decision alternatives together with the importance factors and their relative bearing on the decision. (One of my own back burner projects.)

To assist in following the program flow, here is a description of the main variables.

NI Number of Decision Alternatives
 L\$() Decision Alternative (Array)
 NF Number of Importance Factors
 F\$() Importance Factor (Array)
 V() Relative Value of Each Importance Factor (Array)
 A Pointer to Most Important Factor
 C(,) Two Dimensional Array of Relative Value of Each Alternative With Respect to Importance Factor
 D() Array of Each Alternative's Value
 DF Difference in Value Between Best Two Alternatives
 MX Maximum Value of All Alternatives
 Z() Relative Ranking of Alternatives

QB MONITOR ~ QB-99'er NEWSLETTER

```
100 ! ***** DECISIONS *****
    *   Translated By   *
    *   Ed Machonis    *
    * QB99ers Bayside NY *
    *****
```

```
110 ! ORIGINALLY WRITTEN FOR
    THE PET BY PHIL FELDMAN AND
    TOM RUGG-"DECIDE" COPYRIGHT
    1978
```

```
120 DIM L$(10),F$(10),V(10),
    C(10,10),D(10),Z(10):: E$="D
    ONE"
```

```
130 CALL CLEAR :: GOSUB 920
    :: GOSUB 920
```

```
140 !
```

```
150 PRINT : "I CAN HELP YOU MAKE A
    DECISION. ALL I NEED TO DO IS ASK SOME
    QUESTIONS AND"
```

```
160 PRINT "THEN ANALYZE THE INFORMATION
    YOU GIVE.":TAB(3);RPT$("- ",22)::
    "WHICH ONE BEST DESCRIBES THE TYPE OF
    DECISION FACING YOU?"
```

```
170 PRINT : " 1) CHOOSING AN ITEM FROM
    VARIOUS ALTERNATIVES."
```

```
180 PRINT : " 2) CHOOSING A COURSE OF
    ACTION FROM VARIOUS ALTERNATIVES."
```

```
190 PRINT : " 3) MAKING A "YES" OR
    "NO" DECISION."
```

```
200 INPUT "WHICH ONE (1,2, OR 3)?":T
    :: IF T<1 OR T>3 OR T<>INT(T) THEN 200
```

```
210 GOSUB 920 :: GOSUB 920
```

```
220 ON T GOTO 230,240,250
```

```
230 PRINT :: INPUT "WHAT TYPE OF ITEM
    MUST YOU DECIDE UPON?":T$ :: IF T$=""
    THEN 230 ELSE PRINT :: GOTO 260
```

```
240 T$="COURSE OF ACTION" :: GOTO 260
```

```
250 T$=""YES" OR "NO" :: NI=2 :: L$(1)=""
    DECIDING YES" :: L$(2)=""DECIDING NO" ::
    GOTO 370
```

```
260 GOSUB 920 :: NI=0 :: PRINT : "I NEED
    A LIST OF EACH":T$;" UNDER CONSIDERATION."
    ::
```

```
270 PRINT "INPUT THEM ONE AT A TIME IN
    RESPONSE TO EACH QUESTION MARK.":T$;"
    THE ORDER IN WHICH YOU"
```

```
280 PRINT "INPUT THEM DOES NOT MATTER.":
    T$;" TYPE THE WORD "DONE" WHEN YOU ARE
    DONE." ::
```

```
290 NI=NI+1
```

```
300 INPUT L$(NI):: IF L$(NI)="" THEN PRINT
    "TYPE "DONE" WHEN YOU ARE DONE ELSE
    ENTER A":T$ :: GOTO 300
```

```
310 IF NI=10 THEN 320 ELSE IF L$(NI)<>E$
    THEN 290 ELSE 330
```

```
320 PRINT : "MAXIMUM OF 10 FACTORS CAN
    BE EVALUATED" :: GOSUB 930 :: GOTO 340
```

```
330 NI=NI-1 :: IF NI>=2 THEN 340 ELSE
    PRINT : "YOU MUST HAVE AT LEAST TWO
    CHOICES.":T$;" TRY AGAIN." :: GOSUB 930
    :: GOTO 260
```

```
340 GOSUB 920 :: PRINT : "O.K. HERE'S
    THE LIST YOU'VE GIVEN ME":T$ ::
```

```
350 FOR J=1 TO NI :: PRINT J;"":L$(J)
    :: NEXT J :: PRINT
```

```
360 INPUT "IS THIS LIST CORRECT? (Y OR
    N)":R$ :: IF R$="Y" THEN 370 ELSE
    PRINT : "THE LIST MUST BE RE-ENT
```

```
ERED." :: GOSUB 930 :: GOTO 260
```

```
370 GOSUB 920
```

```
380 IF T<3 THEN PRINT : "NOW, THINK OF
    THE DIFFERENT FACTORS THAT ARE
    IMPORTANT TO YOU IN CHOOSING THE BEST":
    T$;"." :: GOTO 400
```

```
390 PRINT : "NOW, THINK OF THE DIFFERENT
    FACTORS THAT ARE IMPORTANT TO YOU IN
    DECIDING "YES" OR "NO"."
```

```
400 PRINT : "INPUT THEM ONE AT A TIME IN
    RESPONSE TO EACH QUESTION MARK.":T$;"
    TYPE THE WORD "DONE" WHEN YOU ARE
    DONE." :: NF=0
```

```
410 NF=NF+1
```

```
420 INPUT F$(NF):: IF F$(NF)="" THEN PRINT
    "TYPE THE WORD "DONE" WHEN YOU ARE
    DONE ELSE ENTER A FACTOR" :: GOTO 420
```

```
430 IF NF=10 THEN 440 ELSE IF F$(NF)<>E$
    THEN 410 ELSE 450
```

```
440 PRINT : "MAXIMUM OF 10 FACTORS CAN
    BE EVALUATED!" :: GOSUB 930 :: GOTO 460
```

```
450 NF=NF-1 :: IF NF<1 THEN PRINT : "YOU
    MUST HAVE AT LEAST ONE! REDO IT!" ::
    GOSUB 930 :: GOTO 380
```

```
460 GOSUB 920 :: PRINT : "HERE'S THE
    LIST OF FACTORS YOU GAVE ME":T$ ::
```

```
470 FOR J=1 TO NF :: PRINT J;"":F$(J)
    :: NEXT J
```

```
480 PRINT : "DECIDE WHICH FACTOR ON THE
    LIST IS THE MOST IMPORTANT AND INPUT
    ITS NUMBER.":T$(TYPE "0" IF THE LIST
    NEEDS CHANGING.)"
```

QB MONITOR ~ QB-99'er NEWSLETTER

DECISIONS.....Page 2

```
490 INPUT A : A=INT(A) : IF
  A=0 THEN 370 ELSE IF A>NF T
  HEN 460 ELSE GOSUB 920
```

```
500 IF NF=1 THEN 560
```

```
510 PRINT : "NOW LET'S SUPPOS
  E WE HAVE A SCALE OF IMPORTA
  NCE RANGING FROM 0-10."
```

```
520 PRINT : "WE'LL GIVE ";F$(
  A):"A VALUE OF 10 SINCE ";F$(
  A):"WAS RATED THE MOST IMPO
  RTANT"
```

```
530 PRINT : "ON THIS SCALE, W
  HAT VALUE OF IMPORTANCE WOULD
  THE OTHER FACTORS HAVE?";
  :
```

```
540 FOR J=1 TO NF : IF J=A
  THEN 555 ELSE PRINT
```

```
550 INPUT F$(J)&": ";V(J):
  IF V(J)<0 OR V(J)>10 THEN PR
  INT : "IMPOSSIBLE VALUE - TRY
  AGAIN": : GOTO 550
```

```
555 NEXT J
```

```
560 V(A)=10 : Q=0
```

```
570 FOR J=1 TO NF : Q=Q+V(J)
  : NEXT J
```

```
580 FOR J=1 TO NF : V(J)=V(
  J)/Q : NEXT J : GOSUB 920
```

```
590 IF T<3 THEN PRINT : "EACH
  ";T$ ELSE PRINT : "DECIDING
  YES OR DECIDING NO"
```

```
600 PRINT "MUST NOW BE COMPA
  RED WITH EACH IMPORTANCE F
  ACTOR.": : "WE'LL CONSIDER EA
  CH FACTOR SEPARATELY AND TH
  EN RATE"
```

```
610 IF T<3 THEN PRINT "EACH
  ";T$ ELSE PRINT "DECIDING YE
  S OR DECIDING NO"
```

```
620 PRINT "IN TERMS OF THAT
  FACTOR ONLY": : "LET'S GIVE "
```

```
;L$(1):"A VALUE OF 10 ON EVE
  RY SCALE": : :
```

```
630 IF T<3 THEN PRINT "THEN
  EVERY OTHER ";T$ ELSE PRINT
  "THEN DECIDING NO"
```

```
640 PRINT "WILL BE ASSIGNED
  A VALUE HIGHER OR LOWER T
  HAN 10. THIS VALUE DEPEND
  S ON HOW"
```

```
650 PRINT "MUCH YOU THINK IT
  IS BETTER OR WORSE THAN ";L
  $(1): : "PRESS ANY KEY TO CON
  TINUE."
```

```
660 CALL KEY(O,K,S) : IF S=0
  THEN 660
```

```
670 FOR J=1 TO NF
```

```
680 PRINT : TAB(3);RPT$("-",2
  2): : "CONSIDERING ONLY ";F$(
  J);": "AND ASSIGNING 10 TO
  ";L$(1);": "WHAT VALUE WOU
  LD YOU GIVE TO": : :
```

```
690 FOR K=2 TO NI
```

```
700 INPUT L$(K)&": ";C(K,J):
  : PRINT : IF C(K,J)<0 THEN
  PRINT "-NEGATIVE VALUES NOT
  LEGAL-" : GOTO 700
```

```
710 NEXT K : PRINT : C(1,J)
  )=10 : NEXT J
```

```
720 FOR J=1 TO NF : Q=0 :
  FOR K=1 TO NI : Q=Q+C(K,J):
  : NEXT K
```

```
730 FOR K=1 TO NI : C(K,J)=
  C(K,J)/Q : NEXT K : NEXT J
```

```
740 FOR K=1 TO NI : D(K)=0
  : FOR J=1 TO NF : D(K)=D(K)
  +C(K,J)*V(J) : NEXT J : NE
  XT K
```

```
750 MX=0 : FOR K=1 TO NI :
  IF D(K)>MX THEN MX=D(K)
```

```
760 NEXT K : FOR K=1 TO NI
  : D(K)=D(K)*100/MX : NEXT
  K
```

```
770 FOR K=1 TO NI : Z(K)=K
  : NEXT K : NM=NI-1
```

```
780 FOR K=1 TO NI : FOR J=1
  TO NM : N1=Z(J) : N2=Z(J+1)
  : IF D(N1)>D(N2) THEN 800
```

```
790 Z(J+1)=N1 : Z(J)=N2
```

```
800 NEXT J : NEXT K : J1=Z
  (1) : J2=Z(2) : DF=D(J1)-D(J
  2) : GOSUB 920
```

```
810 PRINT : L$(J1)
```

```
820 PRINT "COMES OUT BEST."
```

```
830 IF DF<5 THEN PRINT "BUT
  IT'S VERY CLOSE." : GOTO 87
  0
```

```
840 IF DF<10 THEN PRINT "BUT
  IT'S FAIRLY CLOSE." : GOTO
  870
```

```
850 IF DF<20 THEN PRINT "BY
  A FAIR AMOUNT." : GOTO 870
```

```
860 PRINT "QUITE DECISIVELY!"
  :
```

```
870 PRINT : "HERE IS THE FINA
  L LIST:"
```

```
880 PRINT : L$(J1); " HAS BEEN
  "; "GIVEN A VALUE OF 100 AND
  THE OTHERS RATED ACCORDINGLY.
  "
```

```
890 GOSUB 930 : PRINT TAB(3)
  ;RPT$("-",22)
```

```
900 FOR J=1 TO NI : Q=Z(J):
  : PRINT : TAB(4-LEN(STR$(INT(
  D(Q)))));INT(D(Q));TAB(8);L$(
  Q) : NEXT J
```

```
910 END
```

```
920 FOR J=1 TO 6 : PRINT : T
  AB(3);"? ? ? DECISIONS ? ? ?
  " : NEXT J : RETURN
```

```
930 FOR J=1 TO 1000 : NEXT
  J : RETURN
```

CATALABEL REVISITED
CATALABEL REVISITED

By Ed Nachonis

The double header is not a mistake nor is the computer/printer suffering a case of hic-cups. There are two villains in this piece. An Editor who is quick on the draw and a programmer who can't learn to let go.

I had no sooner sent off my copy to our editor last month than a second look at the programs indicated what seemed to me a better program structure and a nicer label. A quick call to the editor and new copy in the mail proved of no avail. The newsletter was late (So what else is new?) and waited on no man.

I think the improvement is worth another REVISIT and will give us another opportunity to prove Babbage's Law of Programming Diversity: "There's More Than One Way to Skin a Cat!" (For additional info, see the frontispiece in our Dec 88 issue.)

The previous versions required separate input statements and variables for each of the three programs displayed on a line of the label. Separate print commands were also issued. In the new versions, a loop is used and the variables are placed into, and printed from, an array. (This is an improvement?)

The big change is that CATALABEL 2 can now display the record length. As I had stated last month, I was not happy giving up this information (I couldn't distinguish between a letter to Aunt Sophie and a program in Merge format.) and I guess that's what prompted that second look.

In order to list three programs on one line of a mailing label, complete with record lengths, file types are displayed in a different manner. It is the method used in J. Johnson's BOOT program. Capital letters denote FIXED record length and lower case letters

mean VARIABLE length records. "D" = DIS/FIX, "d" = DIS/VAR, "I" = INT/FIX and "i" = INT/VAR.

As we use a wider label with CATALABEL 3, the file type display has not been changed in that program. Once you have typed in CATALABEL 2, only a few changes are required to arrive at CATALABEL 3. Change 59 to 67 in Line 5; change 22 to 34 in Line 6; change the Print Using string in Line 8 (add two number signs and an extra space to the last group); change 24 to 36 in Line 9 and the DATA items in Line 10. Oh yes, change the name in Line 1 and label size in Line 2.

The revised listings are as follows:

```
1 ! *****
  * CATALABEL 2 *
  * By Ed Nachonis *
  * QB99ers Bayside NY *
  *****
```

```
2 ! Prints A 3 Column Disk
   Catalog On A 1" x 3-1/2"
   Mailing Label
```

```
3 OPEN #2:"PI0" :: PRINT #2:
  CHR$(15);CHR$(27);"A";CHR$(9
  ):: FOR I=1 TO 5 :: READ T$(
  I):: NEXT I :: CALL CLEAR
```

```
4 INPUT "READY?";I$ :: OPEN
  #1:"DSK1.",INPUT ,RELATIVE,1
  NTERNAL :: INPUT #1:A$,J,J,K
```

```
5 PRINT #2:CHR$(14);CHR$(27)
  ;"G";A$;CHR$(20);CHR$(27);"H
  ";
  AVAIL";K;":USED"
  ;J-K:RPT$("=",59):: L=6 :: F
  OR I=1 TO 43 :: FOR K=1 TO 3
```

```
6 INPUT #1:B$(K),E(K),S(K),M
  (K):: IF LEN(B$(1))=0 THEN 9
  ELSE L=L+1 :: IF L=22 THEN
  PRINT #2 :: L=1
```

```
7 IF M(K)THEN N$(K)=STR$(M(K
  ))ELSE N$(K)=""
```

```
8 PRINT #2,USING "#####
  ### ### " :B$(K),S(K),T$(AB
```

```
S(E(K)))&M$(K):: NEXT K ::
  PRINT #2 :: NEXT I
```

```
9 CLOSE #1 :: FOR I=1 TO 24-
  L STEP 3 :: PRINT #2 :: NEXT
  I :: CALL CLEAR :: GOTO 4
```

```
10 DATA D,d,I,i,PGM
```

```
1 ! *****
  * CATALABEL 3 *
  * By Ed Nachonis *
  * QB99ers Bayside NY *
  *****
```

```
2 ! Prints A 3 Column Disk
   Catalog On A 1-1/2" x 4"
   Mailing Label
```

```
3 OPEN #2:"PI0" :: PRINT #2:
  CHR$(15);CHR$(27);"A";CHR$(9
  ):: FOR I=1 TO 5 :: READ T$(
  I):: NEXT I :: CALL CLEAR
```

```
4 INPUT "READY?";I$ :: OPEN
  #1:"DSK1.",INPUT ,RELATIVE,1
  NTERNAL :: INPUT #1:A$,J,J,K
```

```
5 PRINT #2:CHR$(14);CHR$(27)
  ;"G";A$;CHR$(20);CHR$(27);"H
  ";
  AVAIL";K;":USED"
  ;J-K:RPT$("=",67):: L=6 :: F
  OR I=1 TO 43 :: FOR K=1 TO 3
```

```
6 INPUT #1:B$(K),E(K),S(K),M
  (K):: IF LEN(B$(1))=0 THEN 9
  ELSE L=L+1 :: IF L=34 THEN
  PRINT #2 :: L=1
```

```
7 IF M(K)THEN N$(K)=STR$(M(K
  ))ELSE N$(K)=""
```

```
8 PRINT #2,USING "#####
  ### ### " :B$(K),S(K),T$(
  ABS(E(K)))&M$(K):: NEXT K
  :: PRINT #2 :: NEXT I
```

```
9 CLOSE #1 :: FOR I=1 TO 36-
  L STEP 3 :: PRINT #2 :: NEXT
  I :: CALL CLEAR :: GOTO 4
```

```
10 DATA D/F,D/V,I/F,I/V,PRGM
```

TIPS FROM THE TIGERCUB

#43

Copyright 1987

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, now reduced to just \$2.00 each, plus \$1.50 per order for cassette or disk and PP&M. Cassette programs will not be available after my present stock of blanks is exhausted.

[Programs have been further reduced to \$1.00 each. ED.]

Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$10 postpaid. Each of these contains either 5 or 6 of my regular \$2 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!

TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION

REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

[Collections have now been reduced to \$5.00 each. ED.]

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloder, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to \$15.00 postpaid.

NUTS & BOLTS NO. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$15 postpaid.

* NUTS & BOLTS #3 is now *
* ready, another full disk *
* of 140 new merge-format *
* utility subprograms, all *
* compatible with the pre- *
* vious. With 11 pages of *
* documentation, \$15 ppd. *

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to \$10 ppd.

TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$10
TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tips and routines from Nos. 25

through 32, \$10 postpaid.
TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also \$10 postpaid.

If you have as much trouble as I do, trying to get the strip labels lined up in the printer, you'll like this one -

```
100 DISPLAY AT(4,7)ERASE ALL
:"TIGERCUB LABELER": : : :
This label maker will allow
:"you to specify different":
"printer codes for each line
"
```

```
110 DISPLAY AT(11,1):"of a 5
-line label.": : : " You may
stop the program": "while lab
els are printing": "by pressi
ng any key, turn"
```

```
120 DISPLAY AT(17,1):"off th
e printer to adjust": "the la
bels, turn it back on," and
press any key to con-": "tin
ue printing."
```

```
130 DISPLAY AT(23,1):"Printe
r designation?": "PID" :: ACC
EPT AT(24,1)SIZE(-28)BEEP:PR
$ :: OPEN @1:PR$ :: P$,E$,DS
$,CEN$="Y" :: DW$,I$,SS$,U$=
"N" :: P=1
```

```
140 CALL CHAR(95,"FF")
150 FOR J=1 TO 5 :: CALL KEY
(3,K,S)
```

```
160 DISPLAY AT(2,1)ERASE ALL
:"Line #";J;" - PRINT? "&P$
:: CALL QUERY(2,20,P$):: IF
P$="N" THEN L$(J)=" " :: GOTO
360
```

```
170 IF J>1 THEN DISPLAY AT(4
,1):"Change codes? N" :: CAL
L QUERY(4,15,Q$):: IF Q$="N"
THEN 300
```

```
180 DISPLAY AT(4,1):"Print p
itch? ";P;" (1)pica": " (2)el
ite": " (3)condensed" :: ACCE
PT AT(4,15)SIZE(-1)VALIDATE(
"123"):P
```

```
190 CI=(P=1)*-10+(P=2)*-12+(
P=3)*-17 :: L$(J)=CHR$(27)&"
B"&CHR$(P):: DISPLAY AT(5,1)
:"": ""
```


TIGERCUB TIPS #43 Page 2

```

200 DISPLAY AT(6,1):"Double
width? "&DW$ :: CALL QUERY(6
,15,DW$):: IF DW$="Y" THEN C
I=CI/2 :: L$(J)=L$(J)&CHR$(1
4)ELSE L$(J)=L$(J)&CHR$(20)
210 DISPLAY AT(8,1):"Italics
? "&I$ :: CALL QUERY(8,10,I$
):: IF I$="Y" THEN L$(J)=L$(
J)&CHR$(27)&"4" ELSE L$(J)=L
$(J)&CHR$(27)&"5"
220 DISPLAY AT(10,1):"Supers
cript? "&SS$ :: CALL QUERY(1
0,14,SS$):: IF SS$="Y" THEN
L$(J)=L$(J)&CHR$(27)&CHR$(83
)&CHR$(0)ELSE L$(J)=L$(J)&C
HR$(27)&CHR$(84)
230 IF SS$="Y" THEN 250
240 DISPLAY AT(12,1):"Double
-strike? "&DS$ :: CALL QUERY
(12,16,DS$):: IF DS$="Y" THE
N L$(J)=L$(J)&CHR$(27)&"6" E
LSE L$(J)=L$(J)&CHR$(27)&"H"
250 IF P<>1 OR SS$="Y" THEN
270 :: DISPLAY AT(14,1):"Emp
hasized? "&E$ :: CALL QUERY(
14,13,E$)
260 IF E$="Y" THEN L$(J)=L$(
J)&CHR$(27)&"E" ELSE L$(J)=L
$(J)&CHR$(27)&"F"
270 DISPLAY AT(16,1):"Underl
ine? "&U$ :: CALL QUERY(16,1
2,U$)
280 IF U$="N" THEN L$(J)=L$(
J)&CHR$(27)&CHR$(45)&CHR$(0)
290 DISPLAY AT(18,1):"Center
text? Y" :: CALL QUERY(18,1
4,CEN$)
300 DISPLAY AT(18,1):"Type 1
ine";J;. Enter each":"scree
n line, enter again":"when d
one." :: DISPLAY AT(22,1):RP
T$("_",INT(CI*3.5)): R=21 :
: CALL KEY(5,K,S)
310 ACCEPT AT(R,1):M$ :: IF
M$="" THEN 320 :: A$=A$&M$ :
: R=R+1 :: GOTO 310
320 IF LEN(A$)>INT(CI*3.5)TH
EN DISPLAY AT(16,1):"LINE TO
O LONG!" :: CALL SOUND(300,1
10,0,-4,0):: A$="" :: R=21 :
: GOTO 310

```

```

330 L=LEN(A$):: IF U$="Y" TH
EN A$=CHR$(27)&CHR$(45)&CHR$
(1)&A$&CHR$(27)&CHR$(45)&CHR
$(0)
340 IF CEN$="Y" THEN A$=RPT$
(" ",(INT(CI*3.5)-L)/2)&A$
350 L$(J)=L$(J)&A$ :: A$=""
360 NEXT J
370 DISPLAY AT(12,1)ERASE AL
L:"Print how many?" :: ACCEP
T AT(12,17):N
380 FOR J=1 TO M :: FOR K=1
TO 6 :: PRINT #1:L$(K):: NEX
T K
390 CALL KEY(0,K,S):: IF S=0
THEN 410 ELSE CLOSE #1
400 CALL KEY(0,K1,S1):: IF S
1<1 THEN 400 ELSE OPEN #1:PR
$
410 NEXT J
420 DISPLAY AT(12,8)ERASE AL
L:"Another?" :: CALL QUERY(1
2,17,Q$):: IF Q$="N" THEN ST
OP ELSE 150
430 SUB QUERY(R,C,W$):: ACCE
PT AT(R,C)SIZE(-1)VALIDATE("
YN")BEEP:Q$ :: SUBEND

```

More peculiarities of the TI computer -

```

90 CALL CLEAR :: PRINT TAB(7
);"SPRITE PUZZLE #1":
from Tigercub"
100 PRINT "A non-existent sp
rite can be": "created by CAL
L MOTION.": "It apparently
starts in"
110 PRINT "dot-row 1, dot-co
lumn 1, and": "has color 1. b
ut its pattern": "is not that
of any ASCII!"
120 !by Jim Peterson
130 FOR CH=0 TO 255 :: PRINT
CHR$(CH):: NEXT CH
135 PRINT "CALL MOTION(#1,5,
5):: CALL COLOR(#1,16):: CAL
L MAGNIFY(4)"
140 CALL MOTION(#1,5,5):: CA
LL COLOR(#1,16):: CALL MAGNI
FY(4)
150 GOTO 150

```

And another. -

```

100 DISPLAY AT(3,5)ERASE ALL
:"SPRITE PUZZLE #2":
from Tigercub"
110 DISPLAY AT(7,1):"Non-exi
stent sprites can be": "creat
ed by CALL COLOR.": "Their
existence can be con-"
120 DISPLAY AT(11,1):"firmed
by CALL COINC, but": "CALL P
OSITION reports that": "they
have no position!"
130 CALL COLOR(#1,16):: CALL
COLOR(#2,16)
140 CALL COINC(#1,#2,1,X)::
DISPLAY AT(15,1):"COINC #1,#
2=";X :: CALL POSITION(#1,X,
Y)
150 CALL POSITION(#1,X,Y)::
DISPLAY AT(17,1):"POSITION #
1=";X;Y
160 CALL POSITION(#2,X,Y)::
DISPLAY AT(19,1):"POSITION #
2=";X;Y
170 IF FLAG=1 THEN 140 :: FL
AG=1
180 DISPLAY AT(21,1):"PRESS
ANY KEY"
190 CALL KEY(0,K,S):: IF S=0
THEN DISPLAY AT(21,1):"pres
s any key" :: GOTO 180
200 DISPLAY AT(21,1):"Until
they're set in motion!"
210 CALL MOTION(#1,5,5):: CA
LL MOTION(#2,-5,-5):: GOTO 1
50

If you have the Terminal
Emulator II, Speech Synthe-
sizer, and a pre-schooler in
the house, this will help
him to grasp the idea of
spelling as well as letter
recognition and keyboard
familiarization-

100 REM PRE-SPELLER BY JIM
PETERSON
110 REM TI BASIC WITH TERMI
NAL EMULATOR II AND SPEECH S
YNTHESIZER
120 CALL CLEAR

```