1997

A comparison of the programming languages of the TI-82 and Casio CFX-9850G graphing calculators

Laura M. Bristol
University of Dayton

A COMPARISON OF THE PROGRAMMING LANGUAGES OF THE TI-82 AND CASIO CFX-9850G GRAPHING CALCULATORS

MASTER'S PROJECT

Submitted to the School of Education

University of Dayton, in Partial Fulfillment

of the Requirements for the Degree

Master of Arts in Teaching

by

Laura M. Bristol

School of Education

UNIVERSITY OF DAYTON

Dayton, Ohio

May 1997

Approved by:

Official Advisor

TABLE OF CONTENTS

INTRODUCTION

Not long after birth, children of the soon-to-be 21st century begin developing an affinity for screens. Before their first day of kindergarten, children are entertained and taught by television, computers, and movies. For these kids, technology is a natural medium for interaction. It makes sense for teachers and schools to try to tap into technology as much as they can to meet instructional objectives. In junior high and high school math classes, hand-held graphing calculators about the size of a Game Boy have become a comparatively inexpensive way to access mathematics software. The programming capabilities of these calculators are an especially powerful resource for educators. By using and creating graphing calculator programs with their students, teachers help students focus on understanding algorithms and logical processes. Once students have a knowledge of BASIC programming, they can begin to create useful and sometimes very sophisticated programs.

While Texas Instruments is not the only company marketing graphing calculators, it is the creator of the graphing calculators now most prevalent in schools, the TI-81 and TI-82. For the TI-82, two-person games like Connect4 and Battleship, as well as programs and lesson plans that teachers have used to teach math concepts, can all be accessed on the Internet. Texas Instruments has also published numerous other books of instructional materials for use with the calculators, and magazines such as Mathematics Teacher include graphing calculator activities for the TI-82 in almost every issue.

For students buying a graphing calculator, and for teachers making decisions about which calculators to purchase as classroom sets, the question arises, "Is the TI-82 calculator the only reasonable choice or might another calculator have additional desirable features that would help explain and illustrate mathematics concepts?" For example, the latest Casio graphing calculator,

the CX-9850G, is equipped to graph functions in three different colors. Its "Home" screen is an icon menu, which includes many of the same options as the TI-82, as well as some that are new and unique. In Dynamic Graphing mode, a function can be entered with a variable in place of a coefficient or a constant, and a range for the variable can be specified. After pressing enter, the calculator displays the graph of the function for each value of the variable, showing the effect of that variable on the graph in a dynamic way. The CFX-9850G can also perform operations with complex numbers and has memory space for 36 lists as well as a special function memory, for those functions most often used. While it is not the purpose of this paper to make judgments about which calculator is better, it would be in the best interests of all if a way could be found to translate those programs which have been found to be most useful from one graphing calculator language to another. This manual, comparing the programming commands of the TI-82 and the Casio CFX-9850G, is a first step in that direction.

The manual is written using the TI-82 menus as a starting place. However, the comparison is meant to work both ways. If no exact match for a TI-82 command exists on the Casio, a workable substitute is offered whenever possible, and vice versa. If no substitute commands were found, the statement "Not available" appears in the chart. The first two columns of the charts compare the commands descriptively, with italicized words written in for the parameters. The symbolic Casio commands are replicated exactly, with one exception: ◁ is used to indicate the Display command. The third and fourth columns offer examples of the way each command might appear within a program, using values and variables for the parameters. The fifth column tells the user, in step-by-step fashion, where to locate each command on the Casio. The menu items above the function keys are given in sequence, with > representing the F6 (next menu) command. Part II of the manual lists several menus which are entirely unique to the CFX-9850G.

The author hopes this manual will be useful in several ways to teachers and students. For those who are familiar with the TI-82 and would like to learn about the CFX-9850G, the manual could serve as a miniature course in "Casio". For students with Casios in classrooms where the TI-82 is used on a regular basis, the manual could serve as a "quiet" translator, saving the teacher from giving calculator instructions in two languages. And, for all those interested in becoming bilingual or multilingual calculator users, this manual will no doubt be a springboard for further discoveries of computer kinship!

TI-82 PRGM: CTL

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:If *condition*: *commandA* *:commands*	*condition* ⇒ *commandA* ↵ *commands* ↵	:If A=X: Disp "RIGHT ON!"	A=X⇒"RIGHT ON!" ↵	PRGM JUMP F3
:If *condition* **:Then:** *commands* **:End**	**If** *condition* ↵ **Then** *commands* ↵ **IfEnd** ↵	:If A>X :Then: Disp "TOO HIGH" :End	If A>X ↵ Then "TOO HIGH" ◁ IfEnd ↵	PRGM COM
:If *condition* **:Then** *:commands* **:Else** *:commands* **:End**	**If** *condition* ↵ **Then** *commands* ↵ **Else** *commands* ↵ **IfEnd** ↵ OR *condition* ⇒ *command* ↵ *condition* ⇒ *command* ↵	:If A > 0 :Then :Disp "A > 0" :Else :Disp "A ≤ 0" :End	If A > 0 ↵ Then "A > 0" ◁ Else "A≤0" ◁ IfEnd ↵ OR A > 0 ⇒ "A > 0" ◁ A ≤ 0 ⇒ "A ≤ 0" ◁	PRGM COM F1 ~ F4 OR PRGM JUMP F3
:For (*variable, begin, end*) *:commands* **:End**	**For** *begin*→*variable* **To** *end* ↵ *commands* ↵ **Next** ↵	:For (A, 1, 10) :Disp A :End	For 1→A To 10 ↵ A ◁ Next ↵	PRGM COM > F1 ~ F4
:For (*variable, begin, end, increment*) *:commands* **:End**	**For** *begin*→*variable* **To** *end* **Step** *increment* ↵ *commands* ↵ **Next** ↵	:For (A, 1, 10, .5) :Disp A :End	For 1→A To 10 Step .5 ↵ A ◁ Next ↵	PRGM COM > F1 ~ F4

◁ : Display command on Casio

TI-82 PRGM: CTL

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:While *condition* *:commands* **:End**	**While** *condition* ↵ *commands* ↵ **WhileEnd** ↵	:While A>0 :Disp "Positive" :A-1→A :End	While A>0 ↵ "Positive" ◁ A-1→A ↵ WhileEnd ↵	PRGM COM >, > F1 ~ F2
:Repeat *condition* *:commands until condition is true* **:End**	**Do** *commands* ↵ **LpWhile** *condition* ↵	:Repeat B>100 :A+2→B :Disp B :End	Do A+2→B ↵ B ◁ LpWhile B≤100 ↵	PRGM COM >, > F3 ~ F4
:Pause	◁ : Auto-pause after display	:DispGraph :Pause	Drawgraph ◁	PRGM F5
:Pause *value*	*value* ◁	:Prompt A :Pause 2A	?→A ↵ 2A ◁	PRGM F5
:Lbl *label* *:commands* **:Goto** *label*	**Lbl** *label* ↵ *commands* ↵ **Goto** *label* ↵	:Prompt A,B :Lbl 1 :Prompt X :Disp AX+B :Goto 1	"A="?→A:"B="?→B ↵ Lbl 1 ↵ "X="?→X ↵ AX+B ◁ Goto 1 ↵	PRGM JUMP F1 ~ F2
:IS>(*variable, value***)** *:command if variable ≤ value* *:command if variable > value*	**Isz** *variable* ↵ *command if variable ≠ 0* ↵ *command if variable = 0* ↵	:1→A: 0→B :Lbl 1 :A + B→B :IS>(A, 10) :Goto 1 :Disp B	-10→A: 0→B ↵ Lbl 1 ↵ -A + B→B ↵ Isz A ↵ Goto 1 ↵ B ◁	PRGM JUMP F4

◁ : Display command on Casio

TI-82 PRGM: CTL

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:DS<(*variable, value***)** *:command if variable* ≥ *value* *:command if variable* < *value*	**Dsz** *variable* ↵ *command if variable* ≠ *0* ↵ *command if variable* = *0* ↵	:100→A: 0→B :Lbl 1 :1/A + B→B :DS<(A, 5) :Goto 1 :Disp B	100→A: 0→B ↵ Lbl 1 ↵ A≤4⇒Goto 2 ↵ 1 ÷ A + B→B ↵ Dsz A ↵ Goto 1 ↵ Lbl2: B◁	PRGM JUMP F5
:Menu(*"title", "text1", label1, "text2", label2,. . .***)**	See example	:Menu("Games", "Connect Four", 1, "Battleship", 2, "Tetris", 3)	"Games"◁ "1: Connect Four"◁ "2: Battleship"◁ "3: Tetris"◁ ? → N ↵ N=1 ⇒Goto 1 ↵ N=2 ⇒Goto 2 ↵ N=3 ⇒Goto 3 ↵	PRGM JUMP F3
:prgm*name* [select from PRGM: EXEC menu or type *name* using keyboard alphabet]	**Prog** *"name"* ↵ [type *name* using keyboard alphabet]	:prgmDEFAULT	Prog "DEFAULT" ↵	PRGM CTL F1
:Return	**Return** ↵	:Return	Return ↵	PRGM CTL F2
:Stop	**Stop** ↵	:Stop	Stop ↵	PRGM CTL F4

◁: Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Input [Displays current graph with free-moving cursor. Stores coordinates to variable X and Y and resumes execution when Enter is pressed.] **:DispGraph** **:Trace** [produces the same effect using Trace to update X and Y. Using Trace, the screen will pan left or right if Xmin or Xmax is exceeded.]	Any graph displayed within a program may be traced. The cursor keys shift the graph left, right, up, or down. When Enter is pressed to resume execution, the current Trace coordinates are stored to X and Y.	:Func :[*set window parameters*] :"4X+3" → Y_1 :Input :Disp X :Disp Y	Y=Type ↵ [*set window parameters*] "4X+3" → Y1 ↵ Drawgraph ◁ X ◁ Y ◁	PRGM > DISP F2
:Input *variable*	? → *variable* ↵	:Input A	? → A ↵	PRGM F4
:Input *"text", variable* [*text*: up to 16 characters]	"*text*" ? → *variable* ↵	:Input "TIME IN SEC", T	"TIME IN SEC" ? → T ↵	PRGM F4
:Prompt *variableA, variableB, . . .*	"*variable* =" ? → *variable* ↵	:Prompt Xmin, Xmax, Ymin, Ymax	"Xmin=" ? → Xmin ↵ "Xmax=" ? → Xmax ↵ "Ymin=" ? → Ymin ↵ "Ymax=" ? → Ymax ↵	PRGM F4
:Disp [Displays the home screen]	Not available			

◁ : Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Disp *"text"*	*"text"* ◁	:Disp "TI-82"	"CASIO CFX-9850G" ◁	ALPHA F2
:Disp *valueA, value B, . . .*	*valueA* ◁ *valueB* ◁	:Disp A, B, C	A ◁ B ◁ C ◁	PRGM F5
:Disp *"text", valueA, valueB, . . .*	*"text"* ◁ *valueA* ◁ *valueB* ◁	:Disp "COORDINATES", X, Y	"COORDINATES" ◁ X ◁ Y ◁	PRGM F5
:Disp *list* :Pause	*list* ◁	:Disp L_1 :Pause	List 1 ◁	PRGM F5
:Disp *matrix* :Pause	*matrix* ◁	:Disp [A] :Pause	Mat A ◁	PRGM F5
:Func/Param/Pol **:DispGraph** :Pause/Trace	Y=Type/ParamType/ r=Type ↵ **DrawGraph** ◁	:Func :[*set window parameters*] :"2X+1" → Y_1 :DispGraph :Pause	Y=Type ↵ [*set window parameters*] "2X+1" → Y1 ↵ DrawGraph ◁	PRGM > DISP F2
:Func/Param/Pol :"*family of curves*" → *function* **:DispGraph** :Pause/Trace	Y=Type/ParamType/ r=Type ↵ "*family of curves*" → *function* ↵ **DrawGraph** ◁	:Pol :[*set window parameters*] :"{1,2,3}sin 2θ" → r_1 :DispGraph :Pause	r=Type ↵ [*set window parameters*] "A sin 2θ,[A=1,2,3]" → r1 ↵ DrawGraph ◁	PRGM > DISP F2

◁ : Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Func/Param/Pol :"*curve with limited domain*" → *function* **:DispGraph** :Pause/Trace [Graphs vertical lines at boundary values]	Y=Type/ParamType/ r=Type ↵ "*curve with limited domain*" → *function* ↵ **DrawGraph** ◁	:Func :"X^2 + 3X - 5 (2≤X) (X≤4)" → Y_1 :DispGraph :Pause	Y=Type ↵ "X^2 + 3X - 5,[2, 4]" → Y1 ↵ DrawGraph ◁	PRGM > DISP F2
:Seq: Time: Connected **:DispGraph** :Pause/Trace	DrawR-Tbl ↵ **DrawR-Con** ◁	:Seq: Time: Connected :[*set window parameters*] :[*set sequence parameters*] :"(2*n*+1)/(1-3*n*)" → U_n :DispGraph :Pause	a_nType ↵ [*set window parameters*] [*set sequence parameters*] "(2n+1) ÷ (1-3n)" → a_n ↵ DispR-Tbl ↵ DrawR-Con ◁	PRGM > DISP R-Tbl F3
See example: This program enters *n* in L_1, U_n in L_2, and Σ U_n in L_3, and then plots *n* vs. U_n using a statistical xyLine graph.	Σ dispOn ↵ DrawR-Tbl ↵ **DrawR Σ -Con** ◁	:ClrList L_1, L_2, L_3 :For(N, 1, 20) :N → L_1(N) :(2N+1)/(1-3N) → L_2(N) :End :L_2(1) → L_3(1) :For(N, 1, 19) :L_2(N+1)+L_3(N) → L_3(N+1) :End :[*set window parameters*] :Plot1(xyLine, L_1, L_3, •) :DispGraph	a_nType ↵ Σ dispOn ↵ [*set window parameters*] [*set sequence parameters*] 1 → R Start ↵ 20 → R End ↵ "(2n+1) ÷ (1-3n)" → a_n ↵ DispR-Tbl ↵ DrawR Σ -Con ◁	PRGM > DISP R-Tbl F4

◁ : Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Seq: Time: Dot **:DispGraph** :Pause/Trace	DrawR-Tbl ↵ **DrawR-Plt** ◁	:Seq: Time: Dot :[*set window parameters*] :[*set sequence parameters*] :"3U_{n-1}+2" → U_n :DispGraph :Pause	a_{n+1}Type ↵ [*set window parameters*] [*set sequence parameters*] "3a_n +2" → a_{n+1} ↵ DispR-Tbl ↵ DrawR-Plt ◁	PRGM > DISP R-Tbl F5
See example: This program enters n in L_1, U_n in L_2, and ΣU_n in L_3, and then plots n vs. U_n using a statistical scatter plot.	Σ dispOn ↵ DrawR-Tbl ↵ **DrawR Σ -Plt** ◁	:Seq: PlotsOff :ClrList L_1, L_2, L_3 :"3U_{n-1}+2" → U_n :1 → nMin: 5 → nMax :0 → nStart: 1 → U_nStart :nMin → L_1(1) :U_n(U_nStart) → L_2(1) :For(N, nMin+1, nMax) :N → L_1(N) :U_n(N) → L_2(N): End :L_2(1) + U_nStart → L_3(1) :For(N, nMin, nMax-1) :L_2(N+1) + L_3(N) → L_3(N+1): End :[*set window parameters*] :FnOff 1 :Plot1(Scatter, L_1, L_3, •) :DispGraph	a_{n+1}Type ↵ Σ dispOn ↵ [*set window parameters*] "3a_n +2" → a_{n+1} ↵ 1 → R Start ↵ 5 → R End ↵ 1 → a_0 ↵ DispR-Tbl ↵ DrawR Σ -Plt ◁	PRGM > DISP R-Tbl F6

◁ : Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Seq: Web: Connected **:DispGraph** :Trace [press > cursor to draw web lines on graph]	**DrawWeb** *seq name, n* ◁ [*n* = # of web lines, default value is 20]	:Seq: Web: Connected :[*set window parameters*] :[*set sequence parameters*] :"2.9 U_{n-1} (1-U_{n-1}) " → U_n :DispGraph :Trace	a_{n+1}Type ↵ [*set window parameters*] [*set sequence parameters*] "2.9 a_n (1-a_n) " → a_{n+1} ↵ DrawWeb a_{n+1}, 30 ◁	PRGM > DISP R-Tbl F2
See example: This program completes one round of dynamic graphing.	**DrawDyna** ◁	:[*set window parameters*] :Func :"AX+1" → Y_1: FnOff 1 :"(abs (A-5))X+1" → Y_2 :FnOff 2 :FnOn 1 :For(A, 1, 5) :DispGraph :End :FnOff 1: FnOn 2 :For(A, 1, 4) :DispGraph :End	[*set window parameters*] Y=Type ↵ "AX + 1" → Y1 ↵ D Var A ↵ 1 → D Start ↵ 5 → D End ↵ 1 → D pitch ↵ DrawDyna ◁	PRGM > DISP F3
:DispGraph	**DrawStat** ◁	[*set window parameters*] [*enter statistical data*] :PlotsOn 1 :Plot1(Scatter, L_1, L_2, +) :DispGraph	[*set window parameters*] [*enter statistical data*] S-Gph1 DrawOn, Scatter, List1, List2, 1, Cross, Green ↵ DrawStat ◁	PRGM > DISP F1

◁ : Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Func/Param/Pol **:DispTable**	**DispF-Tbl** ◁	:Param [*set table parameters*] :"3T" → X_{1T} :"T^2" → Y_{1T} :DispTable	ParamType ↵ [*set table parameters*] "3T" → Xt1 ↵ "T^2" → Yt1 ↵ DispF-Tbl ◁	PRGM > DISP F4
:Seq **:DispTable**	**DispR-Tbl** ◁	:Seq :[*set table parameters*] :"1/n" → U_n :DispTable	a_nType ↵ [*set table parameters*] "1÷ n" → a_n ↵ DispR-Tbl ◁	PRGM > DISP F5
Not available	**DrawFTG-Con** ◁			PRGM > DISP F-Tbl F2
Not available	**DrawFTG-Plt** ◁			PRGM > DISP F-Tbl F3
:Output(*line, column, "text"***)** [*line*: 1~8, *column*: 1~16]	**Locate** *column, line,"text"* ↵ [*column*: 1~21, *line*: 1~7]	:Output(4, 6, "TI-82")	Locate 9, 4, "CASIO" ↵	PRGM > I/O F1

◁ : Display command on Casio

TI-82 PRGM: I/O

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Output(*line, column, value***)** [*line*: 1~8, *column*: 1~16]	**Locate** *column, line, value* ↵ [*column*: 1~21, *line*: 1~7]	:Output(8, 16, A)	Locate 21, 7, A ↵	PRGM > I/O F1
:getKey	**Getkey** ↵	:getKey	Getkey ↵	PRGM > I/O F2
:ClrHome [clears pictures from the Home screen, but does not ClrDraw]	**ClrText** ↵ **ClrGraph** ↵ **Cls** ↵	:RecallPic Pic1: Pause :ClrHome: ClrDraw :FnOn 1 :DispGraph: Pause :ClrHome :Disp "TEXT": Pause :ClrHome	RclPict 1 ◁ Cls ↵ G SelOn 1 ↵ DrawGraph ◁ ClrGraph ↵ "TEXT" ◁ ClrText ↵	PRGM, > CLR F1~F2 Sketch F1
:ClrTable	Not available			
:PrintScreen	Not available			
:Get(*variable***)** [*variable*: variable value, list, list element, matrix, matrix element, Y=var, graph database, picture]	**Receive(***variable, variable,* ...) ↵ [*variable*: variable value, list, matrix, picture]	:Get(L_1) :Get(L_2)	Receive(L_1, L_2) ↵	PRGM > I/O F4
:Send(*variable***)** [*variable*: same as for **Get**]	**Send(***variable, variable,...***)** ↵ [*variable*: variable values, list, matrix]	:Send([A]) :Send([B])	Send(Mat A, Mat B) ↵	PRGM > I/O F3

TI-82 MODE

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Normal	**Norm** ↵	:Normal	Norm ↵	SET UP > DISP F3
:Sci	**Sci** *n* ↵ [*n* =0~9: significant digits]	:Float :Sci OR :Fix 2 :Sci	Sci 0 ↵ OR Sci 3 ↵	SET UP > DISP F2
:Eng	**Eng** ↵	:Eng	Eng ↵	SET UP > DISP F4
:Float	**Norm** ↵	:Float	:Norm ↵	See above
0123456789 :Fix *n*	**Fix** *n* ↵ [*n* =0~9]	:Fix 5	Fix 5 ↵	SET UP > DISP F1
:Radian	**Radian** ↵	:Radian	Radian ↵	SET UP ANGL F2
:Degree	**Deg** ↵	:Degree	Degree ↵	SET UP ANGL F1

TI-82 MODE

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
Not available	**Grad** ↵		Grad ↵	SET UP ANGL F3
:Func	**Y=Type** ↵	:Func	Y=Type ↵	F4 GRPH TYPE F1
Not available as a function type. See **Shade** in DRAW: DRAW menu.	**Y>Type** ↵ **Y<Type** ↵ **Y≥Type** ↵ **Y≤Type** ↵			F4 GRAPH TYPE > F1~F4
:Param	**ParamType** ↵	:Param	ParamType ↵	F4 GRPH TYPE F3
:Pol	**r =Type** ↵	:Pol	r =Type ↵	F4 GRPH TYPE F2
Not available as a function type. See **Vertical** in DRAW: DRAW menu.	**x = cType** ↵			F4 GRPH TYPE F4

TI-82 MODE

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Seq	**a_nType** ↵ **a_{n+1}Type** ↵	:Seq	a_nType ↵ a_{n+1}Type ↵ a_{n+2}Type ↵	F4 RECR TYPE F1 ~ F2
Not available	**a_{n+2}Type** ↵			F4 RECR TYPE F3
:Connected	**G-Connect** ↵ [for non-recursive graphs] **DrawR-Con** ◁ **DrawR Σ -Con** ◁ [for recursive graphs: See DispGraph options in PRGM: I/O for more info]	:Connected	G-Connect ↵	SET UP > DRAW F1
:Dot	**G-Plot** ↵ [for non-recursive graphs] **DrawR-Plt** ◁ **DrawR Σ -Plt** ◁ [for recursive graphs: See DispGraph options in PRGM: I/O for more info]	:Dot	G-Plot ↵	SET UP > DRAW F2
:Sequential	**SimulOff** ↵	:Sequential	SimulOff ↵	SET UP >, > SIML F2

TI-82 MODE

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Simul	**SimulOn** ↵	:Simul	SimulOn ↵	SET UP >, > SIML F1
:Fullscreen	Automatic Fullscreen display			
:Split	Not available in PRGM mode			

TI-82 Window: FORMAT

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Seq **:Time**	**DrawR-Con** ◁ OR **DrawR-Plt** ◁	Plots *n* on x-axis, U_n on y-axis. See DispGraph for examples.	Plots n / n+1/ n+2 on x-axis, a_n/ a_{n+1}/ a_{n+2} on y-axis. See DispGraph for examples.	
:Seq **:Web**	**DrawWeb** ◁	Plots U_{n-1} on x-axis, U_n on y-axis. See DispGraph for examples.	Plots a_{n+1}/ a_{n+2} on x-axis, a_n/ a_{n+1} on y-axis. Draws web lines. See DispGraph for examples.	
:RectGC	[automatic when graphing y =, Param, x=c Type equations]			
:PolarGC	[automatic when graphing r = Type equations]			
:CoordOn	**CoordOn** ↵	:CoordOn	CoordOn ↵	SET UP COOR F1
:CoordOff	**CoordOff** ↵	:CoordOff	CoordOff ↵	SET UP COOR F2
:GridOn	**GridOn** ↵	:GridOn	GridOn ↵	SET UP GRID F1
:GridOff	**GridOff** ↵	:GridOff	GridOff ↵	SET UP GRID F2

TI-82 Window: FORMAT

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:AxesOn	**AxesOn** ↵	:AxesOn	AxesOn ↵	SET UP AXES F1
:AxesOff	**AxesOff** ↵	:AxesOff	AxesOff ↵	SET UP AXES F2
:LabelOn	**LabelOn** ↵	:LabelOn	LabelOn ↵	SET UP LABL F1
:LabelOff	**LabelOff** ↵	:LabelOff	LabelOff ↵	SET UP LABL F1
Not available	**DerivOn** ↵ **DerivOff** ↵			SET UP > DERV F1~F2
Not available	Σ **dispOn** ↵ Σ **dispOff** ↵			SET UP >, >, > Σ DSP F1~F2
Not available	**FuncOn** ↵ **FuncOff** ↵ [displays function when graph is traced]			SET UP >, > FUNC F1~F2

TI-82 Window: FORMAT

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
See example	**BG-None** ↵	:ClrDraw	**BG-None** ↵	SET UP > BACK F2
See example	**BG-Pict** # ↵	:RecallPic Pic 1 [must be used with every DispGraph command, when background is desired]	BG-Pict 1 ↵	SET UP > BACK F2
See example	**S-WindAuto** ↵	:ZoomStat	S-WindAuto ↵ DrawStat ◁	SET UP >, > S-WIN F1
See example	**S-WindMan** ↵	:[*set window parameters*] :DispGraph	S-WindMan ↵ [*set window parameters*] ↵ DrawStat ◁	SET UP >, > S-WIN F2
Not available	**LocusOn** ↵ **LocusOff** ↵			SET UP >, > LOCS F1~F2

TI-82 TblSet

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
IndpntAsk	Not available			
IndpntAuto	All tables automatically display the values of the independent variable.			
DependAsk	Not available			
DependAuto	All tables automatically display the values of the dependent variable.			

TI-82 ZOOM: ZOOM

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:ZBox	Not available in PRGM mode			
:*value* → Xfact :*value* → Yfact **:Zoom In**	**Factor** *xfact, yfact* ↵	:2 → Xfact :2 → Yfact :Zoom In	Factor 2, 2 ↵ Drawgraph ◁	ZOOM F1
:*value* → Xfact :*value* → Yfact **:Zoom Out**	**Factor** 1 ÷ *xfact*, 1 ÷ *yfact* ↵	:2 → Xfact :2 → Yfact :Zoom Out	Factor 1 ÷ 2, 1 ÷ 2 ↵ Drawgraph ◁	ZOOM F1
:ZDecimal [preset values: -4.7, 4.7, 1, -3.1, 3.1, 1] OR [Any window values such that Δ X=.1 and Δ Y=.1]	**ViewWindow** -6.3, 6.3, 1, -3.1, 3.1, 1 ↵ [Any window values such that Δ X=.1 and Δ Y=.1]	:ZDecimal	ViewWindow -6.3, 6.3, 1, -3.1, 3.1, 1 ↵ Drawgraph ◁	V-Win F1
:ZSquare [zooms to midpoint of current graph and adjusts existing Xmin, Xmax, and Xscl so that Δ X= Δ Y]	**ViewWindow** -63, 63, 10, -31, 31, 10 ↵ [Any window values such that Δ X= Δ Y, or with Xmax - Xmin = 2* (Ymax - Ymin)]	:ZSquare	ViewWindow -3, 17, 1, -5, 5, 1 ↵ Drawgraph ◁	V-Win F1
:Zstandard [preset values: -10, 10, 1, -10, 10, 1]	**ViewWindow** -10, 10, 1, -10, 10, 1 ↵	:ZStandard	ViewWindow -10, 10, 1, -10, 10, 1 ↵ Drawgraph ◁	V-Win F1

TI-82: Δ X = (Xmax - Xmin) / 94
Δ Y = (Ymax - Ymin) / 62

Casio: Δ X = (Xmax - Xmin) / 126
Casio: Δ Y = (Ymax - Ymin) / 63

TI-82 ZOOM: ZOOM

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Deg **:ZTrig** [preset values: -352.5, 352.5, 90, -4, 4, 1]	Deg ↵ **ViewWindow** -540, 540, 90, -1.6, 1.6, .5 ↵ [preset values for Casio ZTrig]	:ZTrig	Deg ↵ ViewWindow -540, 540, 90, -1.6, 1.6, .5 ↵ Drawgraph ◁	V-Win F1
:Rad **:ZTrig** [preset values: -(47/24) π, (47/24) π, π/2, -4, 4, 1 are such that $\Delta X = \pi/24$]	Rad ↵ **ViewWindow** -3 π, 3 π, $\pi \div 2$ ↵ [preset values for Casio ZTrig (values for Y settings may be anything)] OR **ViewWindow** -(63 ÷ 24) π, (63 ÷ 24) π, $\pi \div 2$, -4, 4, 1 [settings to match TI-82]	:ZTrig	Rad ↵ ViewWindow -(63 ÷ 24) π, (63 ÷ 24) π, $\pi \div 2$, -4, 4, 1 ↵ Drawgraph ◁	V-Win F1
:ZInteger [zooms to center chosen by cursor, sets Xscl=10 and Yscl=10, $\Delta X=1$, $\Delta Y=1$]	**ViewWindow** -63, 63, 10, -31, 31, 10 ↵ [Any window values such that $\Delta X=1$ and $\Delta Y=1$]	:ZInteger	ViewWindow -63, 63, 10, -31, 31, 10 ↵ Drawgraph ◁	V-Win F1
:ZoomStat	**S-Wind Auto** ↵ **DrawStat** ◁	:PlotsOn 1 :Plot1(Scatter, L_1, L_2, +) :ZoomStat	S-Wind Auto ↵ S-Gph 1 DrawOn, Scatter, List1, List2, 1, Cross ↵ DrawStat ◁	Set Up >, > S-Win F1 PRGM > DISP F1

TI-82 ZOOM: MEMORY

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Zprevious	Not available in PRGM mode			
:Func/ Par/ Pol **:ZoomSto**	**StoV-Win** *n* ↵ [*n*: 1~6]	:Func :ZoomSto	StoV-Win 1 ↵	V-Win F2
:Seq **:ZoomSto**	**StoGMEM** *n* ↵ [*n*: 1~6]	:Seq :ZoomSto	StoGMEM 1 ↵	V-Win F2
:Func/ Par/ Pol **:ZoomRcl**	**RclV-Win** *n* ↵ [*n*: 1~6]	:Func :ZoomRcl	RclV-Win 1 ↵	V-Win F3
:Seq **:ZoomRcl**	**RclGMEM** *n* ↵ [*n*: 1~6]	:Seq :ZoomRcl	RclGMEM 1 ↵	V-Win F3

TI-82 VARS: Window

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
To define window parameters, store values to the variables below.	**ViewWindow** *Xmin, Xmax, Xscl, Ymin, Ymax, Yscl, Tθmin, Tθmax, Tθptch* ↵			V-Window F1
Xmin **Xmax** **Xscl**	**Xmin** **Xmax** **Xscl**	:0 → Xmin: 5 → Xmax :.5 → Xscl	0 → Xmin: 5 → Xmax .5 → Xscl ↵ OR ViewWindow 0, 5, .5 ↵	VARS V-Win X F1~F3 V-Window F1
Ymin **Ymax** **Yscl**	**Ymin** **Ymax** **Yscl**	:-2.5 → Ymin: 2.5 → Ymax :.5 → Yscl	-2.5 → Ymin: 2.5 → Ymax .5 → Yscl ↵ OR ViewWindow Xmin, Xmax, Xscl, -2.5, 2.5, .5 ↵	VARS V-Win Y F1~F3 V-Window F1
Not available	**RightXmin** **RightXmax** **RightXscl**			VARS V-Win R-X F1~F6
Not available	**RightYmin** **RightYmax** **RightYscl**			VARS V-Win R-Y F1~F6

TI-82 VARS: Window

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
Δ X **Δ Y** [distance between pixels: **Δ X** = (Xmax - Xmin)/94 **Δ Y** = (Ymax - Ymin)/62. Storing values to **Δ X** and **Δ Y** recalculates Xmin and Xmax]	Not available as a variable, but can be calculated using the following formulas: **Δ X** = (Xmax - Xmin)/126 **Δ Y** = (Ymax - Ymin)/62	:-10→Xmin: 1→Xscl :-10→Ymin: 1→Yscl :.5→Δ X: .5→Δ Y :"2X + 1"→Y_1 :DispGraph :Trace	-10→Xmin: 1→Xscl ↵ -10→Ymin: 1→Yscl ↵ .5→X: .5→Y ↵ 126X + Xmin→Xmax ↵ 62X + Ymin→Ymax ↵ "2X + 1"→Y1 ↵ DrawGraph ◁	
To define Zoom Factors, store values to XFact and YFact.	**Factor** *xfct, yfact* ↵			Zoom F2
X Fact **Y Fact**	**Xfct** **Yfct**	:Prompt Xfact :Prompt Yfact :Zoom In	"Xfct" ?→Xfct ↵ "Yfct" ?→Yfct ↵ Factor Xfct, Yfct ↵ DrawGraph ◁	VARS FACT F1~F2 Zoom F2
Tmin **Tmax** **Tstep** [standard: 0, 2π, π/24]	Paramtype ↵ **Tθmin** **Tθmax** **Tθptch** [standard: 0, 2π, π/50]	:0→Tmin: 3→Tmax :.01→Tstep	Paramtype ↵ 0→ Tθmin: 3→Tθmax ↵ .01→Tθptch ↵ OR ViewWindow Xmax, Xmin, Xscl, Ymax, Ymin, Yscl, 0, 3, .01 ↵	VARS V-Win T, θ F1~F3 V-Window F1

TI-82 VARS: Window

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
θmin **θmax** **θstep** [standard: 0, 2π, π/24]	r=Type↵ **Tθmin** **Tθmax** **Tθptch** [standard: 0, 2π, π/50]	:0→θmin: 3π→θmax :π/24→θstep	r=Type↵ 0→Tθmin 3π→Tθmax↵ π÷24→Tθptch↵ OR ViewWindow Xmax, Xmin, Xscl, Ymax, Ymin, Yscl, 0, 3π, π÷24↵	VARS V-Win T, θ F1~F3 V- Window F1
Not available	**RightTθmin** **RightTθmax** **RightTθptch**			VARS V-Win R-T, θ F1~F3
U_nStart **V_nStart** ***n*Start:** 0→*n*Start 1→*n*Start ***n*Min** ***n*Max**	**a_0, a_1, a_2** **b_0, b_1, b_2** implicit in selection of **a_0 / a_1** **R Start** **R End**	:"3U_{n-1} +2" →U_n :0→*n*Start :1→U_nStart :1→*n*Min :5→nMax	"3a_n +2" →a_{n+1}↵ 1→a_0↵ 1→R Start↵ 5→R End↵	VARS > RECR RANG F1~F4/ > F1~F2
Not available	**anStart** **bnStart** [pointer starting point for WEB graph]			VARS > RECR RANG >, F4~F5

TI-82 VARS: Zoom

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
ZXmin **Zxmax** **ZXscl** **Zymin** **Zymax** **ZYscl**	Y=Type ↵ **StoV-Win** *n* [*n*: 1~6] Stored window values may only be accessed in a group by **RclV-Win** *n* [*n*: 1~6].	:Fund :-25 → Zxmin :25 → ZXmax :5 → ZXscl :-25 → Zymin :25 → ZYmax :5 → ZYscl :[*commands*] :ZoomRcl	Y=Type ↵ ViewWindow -25, 25, 5, -25, 25, 5 ↵ StoV-Win 1 ↵ [*commands*] ↵ RclV-Win 1 ↵	V-Win F2~F3
Ztmin **Ztmax** **Ztstep**	Paramtype ↵ **StoV-Win** *n* ↵ [*n*: 1~6] Stored window values may only be accessed in a group by **RclV-Win** *n* [*n*: 1~6].	:Param :0 → Ztmin: 3 → Ztmax :.02 → Ztstep :[*commands*] :ZoomRcl	ParamType ↵ ViewWindow -25, 25, 5, -25, 25, 5, 0, 3, .02 ↵ StoV-Win 2 ↵ [*commands*] ↵ RclV-Win 2 ↵	V-Win F2~F3
Zθmin **Zθmax** **Zθstep**	r=Type ↵ **StoV-Win** *n* ↵ [*n*: 1~6] Stored window values may only be accessed in a group by **RclV-Win** *n* [*n*: 1~6].	:Pol :0 → Zθmin :6π → Zθmax : π/24 → Zθstep :[*commands*] :ZoomRcl	r=Type ↵ ViewWindow -25, 25, 5, -25, 25, 5, 0, 6π, π/24 ↵ StoV-Win 3 ↵ [*commands*] ↵ RclV-Win 3 ↵	V-Win F2~F3

TI-82 VARS: Zoom

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
ZU$_n$Start **ZV$_n$Start** **Z*n*Start** **Z*n*Min** **Z*n*Max**	a_nType, a_{n+1}Type, a_{n+2}Type ↵ **StoGMEM** *n* ↵ [*n*: 1~6] Stored window values for sequences may only be accessed in a group by **RclGMEM** *n* [*n*: 1~6].	:Seq :0→ Z*n*Start :1→ ZU$_n$Start :1→ Z*n*Min :1→ Z*n*Max :[*commands*] :ZoomRcl	1→a_0 ↵ 1→Rstart ↵ 5→REnd ↵ StoGMEM 1 ↵ [*commands*] ↵ RclGMEM 1 ↵	F4 GRPH GMEM F1~F2

TI-82 VARS: GDB / Picture

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
GDB1 ~ GDB6 [stores Graph MODE, Window VARS and FORMAT, all functions in Y= list and their selection status]	**StoGMEM 1~6** **RclGMEM 1~6** [stored units of GMEM are labeled by number only and cannot be referred to apart from the above commands] [stores all functions in Graph Function Menu, their types, colors, selection status, and 1 ViewWindow setting]	:-10 → Xmin: 10 → Xmax :1 → Xscl: -10 → Ymin :10 → Ymax: 1 → Yscl :"2X + 1" → Y_1 :"-2X + 1" → Y_2 :FnOn1: FnOff 2 :StoreGDB GDB1 :[*commands*] :Recall GDB GDB1	ViewWindow -10, 10, 1, -10, 10, 1 ↵ "2X + 1" → Y1 ↵ "-2X + 1" → Y2 ↵ G SelOn 1: G SelOff 2 ↵ StoGMEM 1 ↵ [*commands*] ↵ RclGMEM 1 ↵	GRPH GMEM F1~F2
Pic1 ~ Pic6 [includes drawn elements, plotted functions, axes, and tick marks]	**StoPict 1~6** **RclPict 1~6** [stored pictures are labeled by number only and cannot be referred to apart from the above commands] [includes same items as TI-82 pictures]	:-47 → Xmin: 47 → Xmax :10 → Xscl: -31 → Ymin :31 → Ymax: 10 → Yscl :Circle(0, 0, 20) :StorePic Pic1	ViewWindow -63, 63, 10, -31, 31, 10 ↵ Circle 0, 0, 20 ↵ StoPict 1 ↵	OPTN >, > PICT F1~F2

TI-82 VARS: Statistics

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
n	**n**	:1-Var Stats L_1 :Disp "n", n	1-Variable List1 ◁ [scroll with cursor keys to find n]	VARS STAT X F1
$\bar{x}$	$\bar{x}$	:1-Var Stats L_1 :Disp "$\bar{x}$", $\bar{x}$	1-Variable List1 ◁ [scroll with cursor keys to find $\bar{x}$]	VARS STAT X F2
Sx	**xσn-1**	:1-Var Stats L_1 :Disp "Sx", Sx	1-Variable List1 ◁ [scroll with cursor keys to find xσn-1]	VARS STAT X > F1
σx	**xσn**	:1-Var Stats L_1 :Disp "σx", σx	1-Variable List1 ◁ [scroll with cursor keys to find xσn]	VARS STAT X F5
$\bar{y}$	$\bar{y}$	:2-Var Stats L_1, L_2 :Disp "$\bar{y}$", $\bar{y}$	2-Variable List1, List2 ◁ [scroll with cursor keys to find $\bar{y}$]	VARS STAT Y F1
Sy	**yσn-1**	:2-Var Stats L_1, L_2 :Disp "Sy", Sy	2-Variable List1, List2 ◁ [scroll with cursor keys to find yσn-1]	VARS STAT Y > F1

TI-82 VARS: Statistics

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
σy	yσn	:2-Var Stats L_1, L_2 :Disp "σy ", σy	2-Variable List1, List2 ◁ [scroll with cursor keys to find yσn]	VARS STAT Y F5
minX **maxX**	**minX** **maxX**	:2-Var Stats L_1, L_2 :Disp "minX", minX :Disp "maxX", maxX	2-Variable List1, List2 ◁ [scroll with cursor keys to find minX and maxX]	VARS STAT X > F2~F3
minY **maxY**	**minY** **maxY**	:2-Var Stats L_1, L_2 :Disp "minY", minY :Disp "maxY", maxY	2-Variable List1, List2 ◁ [scroll with cursor keys to find minY and maxY]	VARS STAT Y > F2~F3
Σx Σx^2	Σx Σx^2	:2-Var Stats L_1, L_2 :Disp "Σx", Σx :Disp "Σx^2 ", Σx^2	2-Variable List1, List2 ◁ [scroll with cursor keys to find Σx and Σx^2]	VARS STAT X F3~F4
Σy Σy^2 Σxy	Σy Σy^2 Σxy	:2-Var Stats L_1, L_2 :Disp "Σy", Σy :Disp "Σy^2 ", Σy^2 :Disp "Σxy", Σxy	2-Variable List1, List2 ◁ [scroll with cursor keys to find Σy and Σy^2 and Σxy]	VARS STAT Y F3~F4

TI-82 VARS: Statistics

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
a **b** **c** **d** **e**	**a** **b** **c** **d** **e**	:QuartReg L_1, L_2 :"ax^4 + bx^3 + cx^2 + dx + e"$\rightarrow Y_1$:DispGraph	QuartReg List1, List2 ↵ "ax^4 + bx^3 + cx^2 + dx + e"$\rightarrow$Y1 ↵ DrawGraph ◁ OR S-Gph 1 DrawOn, Quart, List1, List2, 1, Orange ↵ DrawStat ◁	VARS STAT GRPH F1~F5
r	**r**	:LinReg(ax+b) L_1, L_2 :Disp "r", r	LinearReg List1, List2 ◁	VARS STAT GRPH > F1
Q_1 **Med** **Q_3**	**Q_1** **Med** **Q_3**	:1-Var Stats L_1 :Disp Q_1, Med, Q_3	1-Variable List1 ◁ [scroll with cursor keys to find Q_1, Med, and Q_3]	VARS STAT GRPH > F2~F4
Not available	**Mod**			VARS STAT GRPH > F5

TI-82 VARS: Statistics

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
x1 **y1** **x2** **y2** **x3** **y3**	**x1** **y1** **x2** **y2** **x3** **y3**	:[*enter statistical data into L_1 and L_2*] :Med-Med L_1, L_2 :{x1, x2, x3} → L_3 :{y1, y2, y3} → L_4 :Plot1(Scatter, L_1, L_2, □) :Plot2(Scatter, L_1, L_2, +) :PlotsOn 1, 2 :"ax + b" → Y_1 :ZoomStat	[*enter statistical data into L_1 and L_2*] ↵ Med-MedLine List1, List2 ↵ {x1, x2, x3} → List3 ↵ {y1, y2, y3} → List4 ↵ S-Gph 1 DrawOn, Scatter, List1, List2, 1, Square ↵ S-Gph 2 DrawOn, Scatter, List3, List4, 1, Cross ↵ "ax + b" → Y1 ↵ S-Wind Auto: DrawStat ◁ DrawGraph ◁	VARS STAT PTS F1~F6

TI-82 VARS: Table

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
TblMin **Δ Tbl**	**VarRange** [In this mode, tables are constructed according to Start/End/pitch settings]			SET UP >, >, > T-VAR F1
Func/ Par/ Pol **TblMin** **Δ Tbl**	**F Start** **F End** **F pitch**	:"$2X^2$" → Y_1 :0 → TblMin :1 → Δ Tbl :DispTable	VarRange ↵ "$2X^2$" → Y1 ↵ 0 → F Start ↵ 10 → F End ↵ 1 → F pitch ↵ DispF-Tbl ◁	VARS > TABL F1~F3
Not available	**F Result** [matrix of table contents]			VARS > TABL F4
Seq **TblMin** *integer* → **Δ Tbl**	**R Start** **R End**	:"1/n" → U_n :1 → TblMin :1 → Δ Tbl :DispTable	VarRange ↵ "1 ÷ n" → a_n ↵ 1 → R Start ↵ 10 → R End ↵ DispF-Tbl ◁	VARS > RECR RANG F1~F2
Not available	**R Result** [matrix of table contents]			VARS > RECR F3
Not available	**D Start** **D End** **D pitch**			VARS DYNA F1~F3

TI-82 VARS: Table

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
list→ **TblInput** [list of the values of the independent variable in the current table beginning with TblMin]	**VarList1 ~ VarList6** [In this mode, tables are constructed according to values for the independent variable stored in the specified list]	:{1, 4, 16, 25, 36} → TblInput :"1/x" → Y_1 :DispTable	{1, 4, 16, 25, 36} → List1 ↵ VarList 6 ↵ "1 ÷ x" → Y1 ↵ DispF-Tbl ◁	SET UP >, >, >
T-VAR LIST F1~F6

TI-82 Y-VARS: Function / Parametric / Polar / Sequence

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
$Y_1 \sim Y_0$	**Y1 ~ Y20**	:"$2X^2 + 5X - 9$" $\rightarrow Y_1$	"$2X^2 + 5X - 9$" $\rightarrow$ Y1 ↵	VARS GRPH F1
$X_{1T}, Y_{1T} \sim X_{6T}, Y_{6T}$	**Xt1, Yt1 ~ Xt20, Yt20**	:"3T" $\rightarrow X_{1T}$:"T^2" $\rightarrow Y_{1T}$	"3T" $\rightarrow$ X1t ↵ "T^2" $\rightarrow$ Y1t ↵	VARS GRPH F3~F4
$r_1 \sim r_6$	**r1 ~ r20**	:"5 sin 2θ" $\rightarrow r_1$	"5 sin 2θ" $\rightarrow$ r1 ↵	VARS GRPH F2
Not available	**x1 ~ x20**			VARS GRPH F5
U_n	a_n a_{n+1} a_{n+2}	:"n /(n+1)" $\rightarrow U_n$:"$3U_{n-1} + 5$" $\rightarrow U_n$	"n ÷ (n+1)" $\rightarrow a_n$ ↵ "$3a_n + 5$" $\rightarrow a_{n+1}$ ↵ "$3a_{n+1} - 2a_n + 5$" $\rightarrow a_{n+2}$ ↵	VARS > RECR FORM F1~F3
V_n	b_n b_{n+1} b_{n+2}	:"n /(n+1)" $\rightarrow V_n$:"$3U_{n-1} + 5$" $\rightarrow V_n$	"n ÷ (n+1)" $\rightarrow b_n$ ↵ "$3b_n + 5$" $\rightarrow b_{n+1}$ ↵ "$3b_{n+1} - 2b_n + 5$" $\rightarrow b_{n+2}$ ↵	VARS > RECR FORM F4~F6

TI-82 Y-VARS: On/Off

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Func/ Param/ Pol **:FnOn** **:FnOn** *function1,* *function2, . . .*	**G SelOn** *function1* ↵ **G SelOn** *function2* ↵	:Func :FnOn 1 :DispGraph	G SelOn 1 ↵ DrawGraph◁	F4 GRPH SEL F1
:Seq **:FnOn** *function1, function2*	**R SelOn** *function1* ↵ **R SelOn** *function2* ↵	:Seq :FnOn 1 :DispGraph	R SelOn 1 ↵ DrawR-Con◁	F4 > RECR SEL+C F1
Not available	**D SelOn** *function1* ↵ **D SelOn** *function2* ↵		D SelOn 1 ↵ DrawDyna◁	F4 DYNA F1
:FnOn *function1,* *function2, . . .*	**T SelOn** *function1* ↵ **T SelOn** *function2* ↵	:FnOn 1 :DispGraph	T SelOn 1 ↵ DispF-Tbl◁	F4 > TABL F1
:Func/ Param/ Pol **:FnOff** **:FnOff** *function1,* *function2, . . .*	**G SelOff** *function1* ↵ **G SelOff** *function2* ↵	:Func :FnOff 1: FnOn 2 :DispGraph	G SelOff 1: G SelOn 2 ↵ DrawGraph◁	F4 GRPH SEL F2
:Seq **:FnOff** *function1, function.*	**R SelOff** *function1* ↵ **R SelOff** *function2* ↵	:Seq :FnOff 1: FnOn 2 :DispGraph	R SelOff 1: R SelOn 2 ↵ DrawR-Con◁	F4 > RECR SEL+C F2

TI-82 Y-VARS: On/Off

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
Not available	**D SelOff** *function1* ↵ **D SelOff** *function2* ↵		D SelOff 1: D SelOn 2 ↵ DrawDyna ◁	F4 DYNA F2
:FnOn *function1,* *function2, . . .*	**T SelOff** *function1* ↵ **T SelOff** *function2* ↵	:FnOff 1: FnOn 2 :DispTable	T SelOff 1: T SelOn 2 ↵ DispF-Tbl ◁	F4 > TABL F2

TI-82 STATPLOT: PLOTS

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Plot#(*type,Xlist,Flist***)** [*type*: Boxplot, Histogram]	**S-Gph#** *DrawOn, type, Xlist,Flist* ↵ **S-Gph#** *DrawOn, type, Xlist,Flist,color* ↵ [*type*: Hist, MedBox, MeanBox, N-Dist, Broken]	:Plot1(Boxplot, L_1, 1) :Plot2(Histogram, L_2, L_3)	S-Gph1 DrawOn, MedBox, List1, 1) ↵ S-Gph2 DrawOn, Hist, List2, List3, Green ↵	F4 STAT GRPH F1~F3 F4 STAT DRAW F1
:Plot#(*type,Xlist,Ylist***)** **:Plot#(***type,Xlist,Ylist, mark***)** [*type*: Scatter, xyLine]	**S-Gph#** *DrawOn, type, Xlist,Ylist,Flist* ↵ **S-Gph#** *DrawOn, type, Xlist,Ylist,Flist,mark* ↵ **S-Gph#** *DrawOn, type, Xlist,Ylist,Flist,mark,color* ↵ [*type*: Scatter, xyLine]	:Plot1(Scatter, L_1, L_2) :Plot2(xyLine, L_3, L_4,□)	S-Gph1 DrawOn, Scatter, List1, List2, 1 ↵ S-Gph2 DrawOn, xyLine, List3, List4, 1, Square ↵	F4 STAT GRPH F1~F3 F4 STAT DRAW F1
:PlotsOff **:PlotsOff** *plot#, plot#,...*	*S-Gph#* **DrawOff** ↵	:PlotsOff OR :PlotsOff 1, 2, 3	S-Gph1 DrawOff ↵ S-Gph2 DrawOff ↵ S-Gph3 DrawOff ↵	F4 STAT DRAW F2
:PlotsOn **:PlotsOn** *plot#, plot#*	*S-Gph#* **DrawOn** ↵	:PlotsOn 1	S-Gph1 DrawOn ↵	F4 STAT DRAW F1

TI-82 STATPLOT: TYPE / MARK

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
Scatter	**Scatter**	:Plot1(Scatter, L_1, L_2)	S-Gph1 DrawOn, Scatter, List1, List2, 1 ↵	F4 STAT GRPH F4
xyLine	**xyLine**	:Plot2(xyLine, L_3, L_4,□)	S-Gph2 DrawOn, xyLine, List3, List4, 1, Square ↵	F4 STAT GRPH F5
Boxplot	**MedBox**	:Plot3(Boxplot, L_1, 1)	S-Gph1 DrawOn, MedBox, List1, 1) ↵	F4 STAT GRPH > F2
Not available	**MeanBox**			F4 STAT GRPH > F3
Histogram	**Hist**	:Plot2(Histogram, L_2, L_3)	S-Gph2 DrawOn, Hist, List2, List3, Green ↵	F4 STAT GRPH > F1

TI-82 STATPLOT: TYPE / MARK

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
Not available	**N-Dist**			F4 STAT GRPH > F4
Not available	**Broken**			F4 STAT GRPH > F5
□	**Square**	:Plot1(xyLine, L_1, L_2, □)	S-Gph1 DrawOn, xyLine, List1, List2, 1, Square ↵	F4 STAT MARK F1
+	**Cross**	:Plot2(Scatter, L_1, L_2, +)	S-Gph2 DrawOn, Scatter, List1, List2, 1, Cross ↵	F4 STAT MARK F2
•	**Dot**	:Plot3(Scatter, L_1, L_2, •)	S-Gph3 DrawOn, Scatter, List1, List2, 1, Dot ↵	F4 STAT MARK F3

TI-82 STAT: EDIT / CALC

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:SortA(*listname***)** **:SortA(***listI,listD,listD,...***)** [*listI*: Independent list] [*listD*: Dependent list]	**SortA(***listname***)** ↵ **SortA(***listI,listD,listD,...***)** ↵ [*listI*: Independent list] [*listD*: Dependent list]	:SortA(L_1):SortA(L_2) :SortA(L_1,L_2)	SortA(List1):SortA(List2) ↵ SortA(List1, List2) ↵	F4 LIST F1
:SortD(*listname***)** **:SortD(***listI,listD,listD,...***)** [*listI*: Independent list] [*listD*: Dependent list]	**SortD(***listname***)** ↵ **SortD(***listI,listD,listD,...***)** ↵ [*listI*: Independent list] [*listD*: Dependent list]	:SortD(L_1):SortD(L_2) :SortD(L_1,L_2)	SortD(List1):SortD(List2) ↵ SortD(List1, List2) ↵	F4 LIST F2
:ClrList *listA,listB,listC,...*	**ClrList** ↵ [Clears all lists]	:ClrList L_1,L_2,L_3,L_4,L_5,L_6	ClrList ↵	PRGM > CLR F3
The stat lists and regression commands below do not display the calculations performed, except if listed as the last command in a program.	When followed by ◁, stat lists and regression calculations are displayed and can be scrolled.			
:1-Var Stats **:1-Var Stats** *Xlist* **:1-Var Stats** *Xlist,Flist*	 **1-Variable** *Xlist* ◁ **1-Variable** *Xlist,Flist* ◁	:1-Var Stats L_1 :[*display desired stat variables*]	1-Variable List1 ◁	F4 STAT CALC F1
:2-Var Stats **:2-Var Stats** *Xlist,Ylist* **:2-Var Stats** *Xlist,Ylist,Flist*	 **2-Variable** *Xlist,Ylist* ◁ **2-Variable** *Xlist,Ylist,Flist* ◁	:2-Var Stats L_1, L_2 :[*display desired stat variables*]	2-Variable List1, List2 ◁	F4 STAT CALC F2

TI-82 STAT: EDIT / CALC

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Med-Med **:Med-Med** *Xlist,Ylist* **:Med-Med** *Xlist,Ylist,Flist*	**Med-MedLine** *Xlist,Ylist* ◁ **Med-MedLine** *Xlist,Ylist, Flist* ◁	:Med-Med L_1, L_2 :Disp a, b, r	Med-MedLine List1, List2 ◁	F4 STAT CALC > F2
:LinReg(ax + b) **:LinReg(ax + b)** *Xlist,Ylist* **:LinReg(ax + b)** *Xlist,Ylist, Flist*	**LinearReg** *Xlist,Ylist* ◁ **LinearReg** *Xlist,Ylist Flist* ◁	:LinReg(ax + b) L_1, L_2 :Disp a, b, r	LinearReg List1, List2 ◁	F4 STAT CALC > F1
:QuadReg **:QuadReg** *Xlist,Ylist* **:QuadReg** *Xlist,Ylist,Flist*	**QuadReg** *Xlist,Ylist* ◁ **QuadReg** *Xlist,Ylist,Flist* ◁	:QuadReg L_1, L_2 :Disp a, b, c, r	QuadReg List1, List2 ◁	F4 STAT CALC > F3
:CubicReg **:CubicReg** *Xlist,Ylist* **:CubicReg** *Xlist,Ylist,Flist*	**CubicReg** *Xlist,Ylist* ◁ **CubicReg** *Xlist,Ylist,Flist* ◁	:CubicReg L_1, L_2 :Disp a, b, c, d, r	CubicReg List1, List2 ◁	F4 STAT CALC > F4
:QuartReg **:QuartReg** *Xlist,Ylist* **:QuartReg** *Xlist,Ylist,Flist*	**QuartReg** *Xlist,Ylist* ◁ **QuartReg** *Xlist,Ylist,Flist* ◁	:QuadReg L_1, L_2 :Disp a, b, c, d, e, r	QuadReg List1, List2 ◁	F4 STAT CALC > F5

TI-82 STAT: EDIT / CALC

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:LinReg(a + bx) **:LinReg(a + bx)** *Xlist,Ylist* **:LinReg(a + bx)** *Xlist,Ylist, Flist*	Not available			
:LnReg **:LnReg** *Xlist,Ylist* **:LnReg** *Xlist,Ylist,Flist*	**LogReg** *Xlist,Ylist* ◁ **LogReg** *Xlist,Ylist,Flist* ◁	:LnReg L_1, L_2 :Disp a, b, r	LogReg List1, List2 ◁	F4 STAT CALC >> F1
:ExpReg **:ExpReg** *Xlist,Ylist* **:ExpReg** *Xlist,Ylist,Flist*	**ExpReg** *Xlist,Ylist* ◁ **ExpReg** *Xlist,Ylist,Flist* ◁	:ExpReg L_1, L_2 :Disp a, b, r	ExpReg List1, List2 ◁	F4 STAT CALC >> F2
:PwrReg **:PwrReg** *Xlist,Ylist* **:PwrReg** *Xlist,Ylist,Flist*	**PowerReg** *Xlist,Ylist* ◁ **PowerReg** *Xlist,Ylist,Flist* ◁	:PwrReg L_1, L_2 :Disp a, b, r	PowerReg List1, List2 ◁	F4 STAT CALC >> F3

TI-82 STAT: EDIT / CALC

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:"*Regression Equation*" → *function* :ZoomStat [*Regression Equation* must be entered using variables in the Statistics: Equation VARS menu]	S-Gph *n* DrawOn, ***Regression Type***, *Xlist, Ylist, Flist, Color* ↵ DrawStat ◁ [*n*: 1~3] [***Regression Type***: Linear, Med-Med, Quad, Cubic, Quart, Log, Exp, Power]	:LinReg(ax+b), L_1, L_2 :"aX+b" → Y_1 :ZoomStat	LinearReg List1, List2 ↵ "aX+b" → Y1 ↵ DrawGraph ◁ OR S Gph 1 DrawOn, Linear, List1, List2, 1, Green ↵ S-Wind Auto: DrawStat ◁	F4 STAT GRPH >,> F1~F5 F4 STAT GRPH >, >, > F1~F3
See example	*y-value* $\hat{x}$ ◁ *x-value* $\hat{y}$ ◁	:LinReg(ax + b) L_1, L_2 :"aX+b" → Y_1 :Input X :Disp Y_1(X) :Input Y :Disp (Y-b)/a	LinearReg List1, List2 ◁ ? → X ↵ X $\hat{y}$ ◁ ? → Y ↵ Y $\hat{x}$ ◁	OPTN STAT F1~F2

TI-82 LIST: OPS / MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:**L_n**(*element #*)	**Listn**[*element #*]	:Disp L_1(5)	List1[5] ◁	OPTN LIST F1
:**SortA**(*listname*) :**SortA**(*listI, listD, listD, ...*) [*listI*: Independent list] [*listD*: Dependent list]	**SortA**(*listname*) ↵ **SortA**(*listI, listD, listD, ...*) ↵ [*listI*: Independent list] [*listD*: Dependent list]	:SortA(L_1):SortA(L_2) :SortA(L_1,L_2)	SortA(List1):SortA(List2) ↵ SortA(List1, List2) ↵	F4 LIST F1
:**SortD**(*listname*) :**SortD**(*listI, listD, listD, ...*) [*listI*: Independent list] [*listD*: Dependent list]	**SortD**(*listname*) ↵ **SortD**(*listI, listD, listD, ...*) ↵ [*listI*: Independent list] [*listD*: Dependent list]	:SortD(L_1):SortD(L_2) :SortD(L_1,L_2)	SortD(List1):SortD(List2) ↵ SortD(List1, List2) ↵	F4 LIST F2
:**dim** *listname*	**dim** *listname* ◁	:Disp dim L_1	dim List1 ◁	OPTN LIST F3
:*length* → **dim** *listname*	See example	:12 → dim L_2	For 1 → A to 12 Step 1 ↵ 0 → List2[A] ↵ Next ↵	See PRGM: CTL menu
:*length* → **dim** *listname* [re-dimensioning list]	See example	:15 → dim L_2	For dim List2 → A to 15 Step 1 ↵ 0 → List2[A] ↵ Next ↵	See PRGM: CTL menu
:[define dimension of list] :**Fill**(*value, listname*)	[define dimension of list] **Fill**(*value, listname*) ↵	:Fill(8, L_5)	Fill(8, List5) ↵	OPTN LIST F4

TI-82 LIST: OPS / MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:seq(*expression, variable, begin, end, increment***)**	**Seq(***expression, variable, begin, end, increment***)** ◁	:seq(N^2, N, 1, 10, 1) → L_1	Seq(N^2, N, 1, 10, 1) → List1 ↵	OPTN LIST F5
:U_n(*nstart, nstop, nstep***)** **:V_n(***nstart, nstop, nstep***)**	Not available			
:min(*list***)** **:min(***listA, listB***)**	**Min(***list***)** ↵ **Min(***listA, listB***)** ↵	:Disp min(L_1) :Disp min(L_1, L_2)	Min(List1) ◁ Min(List1, List2) ◁	OPTN LIST > F1
:max(*list***)** **:max(***listA, listB***)**	**Max(***list***)** ↵ **Max(***listA, listB***)** ↵	:Disp max(L_1) :Disp max(L_1, L_2)	Max(List1) ◁ Max(List1, List2) ◁	OPTN LIST > F2
:mean(*list***)** **:mean(***list, Flist***)**	**Mean(***list***)** ↵ **Mean(***list, Flist***)** ↵	:Disp mean(L_1) :Disp mean(L_1, L_2)	Mean(List1) ◁ Mean(List1, List2) ◁	OPTN LIST > F3
:median(*list***)** **:median(***list, Flist***)**	**Median(***list***)** ↵ **Median(***list, Flist***)** ↵	:Disp median(L_1) :Disp median(L_1, L_2)	Median(List1) ◁ Median(List1, List2) ◁	OPTN LIST > F4
:sum ***list***	**Sum** ***list*** ↵	:sum L_3 → A	Sum List3 → A ↵	OPTN LIST >> F1

TI-82 LIST / OPS / MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:prod *list*	**Prod** *list* ↵	:prod $L_3 \rightarrow$ B	Prod List3 $\rightarrow$ B ↵	OPTN LIST >> F2
See example	**Cuml** *list* ↵	:{2, 3, 6, 5, 4} $\rightarrow L_1$:$L_1(1) \rightarrow L_2(1)$:For (A, 2, dim L_1, 1) :$L_2(A-1) + L_1(A) \rightarrow L_2(A)$:End :Disp L_2	{2, 3, 6, 5, 4} $\rightarrow$ List 1 ↵ Cuml List 1 ◁	OPTN LIST >> F3
See example	**Percent** *list* ↵	:{2, 3, 6, 5, 4} $\rightarrow L_1$:For (A, 1, dim L_1, 1) :(L_1(A)/Sum L_1) * 100 $\rightarrow$ L_2(A) :End :Disp L_2	{2, 3, 6, 5, 4} $\rightarrow$ List 1 ↵ Percent List 1 ◁	OPTN LIST >> F4
See example	**List** $\rightarrow$ **Mat**(*listA, listB, listC, listD,...*) ↵	:{2, 3, 6, 5, 4} $\rightarrow L_1$:{11, 12, 13, 14, 15 } $\rightarrow L_2$:{5, 2} $\rightarrow$ dim [A] :For(B, 1, 5, 1) :L_1(B) $\rightarrow$ [A](B, 1) :L_2(B) $\rightarrow$ [A](B, 2) :End :Disp [A] :Pause	{2, 3, 6, 5, 4} $\rightarrow$ List 1 ↵ {11, 12, 13, 14, 15 } $\rightarrow$ List 2 ↵ List $\rightarrow$ Mat(List 1, List 2) ◁	OPTN LIST F2

TI-82 MATH: MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:*value*>**Frac** :*list*>**Frac** :*matrix*>**Frac**	*value for numerator*_| *value for denominator* ↵	:N/D>Frac :L_1>Frac :[A]>Frac	N_|D ↵	a b/c on keyboard
:*value*>**Dec** :*list*>**Dec** :*matrix*>**Dec**	[expressions automatically displayed as decimals]	:N/D>Dec :L_1>Dec :[A]>Dec	N ÷ D ◁ List 1 ◁ Mat A ◁	PRGM F5
:*value*^{3} :*list*^{3} :*matrix*^{3}	*value*^**3** ↵ *list*^**3** ↵ *matrix*^**3** ↵	:$X^3 \rightarrow Y$:$L_1^3 \rightarrow L_2$:$[A]^3 \rightarrow [A]$	X^3 → Y ↵ List 1^3 → List 2 ↵ Mat A^3 → Mat A ↵	on keyboard
:$\sqrt[3]{}$ *value* :$\sqrt[3]{}$ *list*	$\sqrt[3]{}$ *value* ↵ $\sqrt[3]{}$ *list* ↵	: $\sqrt[3]{}$ (A - B) :$\sqrt[3]{}$ L_1	$\sqrt[3]{}$ (A - B) ↵ $\sqrt[3]{}$ List 1 ↵	SHIFT (on keyboard
:*nth root* $\sqrt[x]{}$ *value* :*nth root* $\sqrt[x]{}$ *list* :*list* $\sqrt[x]{}$ *value* :*listA* $\sqrt[x]{}$ *listB*	*nth root* $\sqrt[x]{}$ *value* ↵ *nth root* $\sqrt[x]{}$ *list* ↵ *list* $\sqrt[x]{}$ *value* ↵ *listA* $\sqrt[x]{}$ *listB* ↵	:N $\sqrt[x]{}$ R :5 $\sqrt[x]{}$ L_1 :L_1 $\sqrt[x]{}$ 100 :L_1 $\sqrt[x]{}$ L_2	N $\sqrt[x]{}$ R ↵ 5 $\sqrt[x]{}$ List 1 ↵ List 1 $\sqrt[x]{}$ 100 ↵ List 1 $\sqrt[x]{}$ List 2 ↵	SHIFT ^ on keyboard
:**fmin**(*expression, variable, lower, upper*)	**Fmin**(*expression in terms of x, lower, upper*) ↵	:fmin(X^2 - 5X + 9, X, 1, 4)	Fmin(X^2 - 5X + 9, 1, 4) ↵	OPTN CALC > F1
:**fmin**(*expression, variable, lower, upper, tolerance*) [*tolerance*: default is 10^{-5}]	**Fmin**(*expression in terms of x, lower, upper, precision*) ↵ [*precision*: 1~9]	:fmin(X^2 - 5X + 9, X, 1, 4, 10^-6)	Fmin(X^2 - 5X + 9, 1, 4, 6) ↵	OPTN CALC > F1

TI-82 MATH: MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:fmax(*expression, variable, lower, upper***)**	**Fmax(***expression in terms of x, lower, upper***)** ↵	:fmax(-X^2 + 6X - 10, X, 2, 4)	Fmax(-X^2 + 6X -10, 1, 4) ↵	OPTN CALC > F2
:fmax(*expression, variable, lower, upper, tolerance***)** [*tolerance*: default is 10^{-5}]	**Fmax(***expression in terms of x, lower, upper, precision***)** ↵ [*precision*: 1~9]	:fmax(-X^2 + 6X - 10, X, 2, 4, 10^-6)	Fmax(-X^2 + 6X -10, 1, 4, 6) ↵	OPTN CALC > F2
:nDeriv(*expression, variable, value***)**	**d/dx(***expression in terms of x, value***)** ↵	:nDeriv(X^4, X, 3)	d/dx(X^4, 3) ↵	OPTN CALC F2
:nDeriv(*expression, variable, value, e***)** [*e*: default is 10^{-3}]	**d/dx(***expression in terms of x, value, e***)** ↵	:nDeriv(X^4, X, 3, 10^-5)	d/dx(X^4, 3, 10^-5) ↵	OPTN CALC F2
Not available	**d^2/dx^2(***expression in terms of x, value, final boundary***)** ↵ [*final boundary*: 1~15]		d^2/dx^2(X^3 + $2X^2$ - 2X +3, -1) ↵	OPTN CALC F3
:fInt(*expression, variable, lower, upper***)**	∫ (*expression in terms of x, lower, upper*) ↵	:fInt(-$2X^2$ -10X + 1, X, -4, -1)	∫ (-$2X^2$ -10X + 1, -4, -1) ↵	OPTN CALC F4
:fInt(*expression, variable, lower, upper, tolerance***)** [*tolerance*: default is 10^{-5}]	∫ (*expression in terms of x, lower, upper, # of divisions*) ↵ [*# of divisions*: 2^n - enter value for n: 1~9]	:fInt(-$2X^2$ -10X + 1, X, -4, -1, 10^-6)	∫ (-$2X^2$ -10X + 1, -4, -1, 5) ↵	OPTN CALC F4

TI-82 MATH: MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:solve(*expression, variable, guess***)**	**Solve(***expression in terms of x, guess***)** ↵	:solve(-2X^2 -10X + 1, X, -4)	Solve(-2X^2 -10X + 1, -4) ↵	OPTN CALC F1
:solve(*expression, variable, guess, {lower, upper}***)**	**Solve(***expression in terms of x, guess, lower, upper***)** ↵	:solve(-2X^2 -10X + 1, X, -4, {-4.5, -3.5})	Solve(-2X^2 -10X + 1, -4, -4.5, -3.5) ↵	OPTN CALC F1
See example	Σ (*expression, variable, begin, end, increment*) ↵	:"n^2 - 3n + 5" → U_n :U_n(2, 6, 1) → L_1 :Disp Sum L_1	Σ (N^2 - 3N + 5, N, 2, 6, 1) ◁	OPTN CALC > F3

TI-82 MATH: NUM

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:round(*value***)** **:round(***value, #decimals***)** **:round(***list***)** **:round(***list, #decimals***)** **:round(***matrix***)** **:round(***matrix, #decimals***)**	**Fix** *n* ↵ *value* ◁ *list* ◁ *matrix* ◁ **Norm** ↵	:Input A :round(A, 3) :Disp A	Fix 3 ↵ ?→A ↵ A◁ Norm ↵	SET UP > DISP F1
:iPart *value* **:iPart** *list* **:iPart** *matrix*	**Int** *value* ↵ **Int** *list* ↵ **Int** *matrix* ↵	:iPart A→B :iPart L_1→L_2 :iPart [A]→[B]	Int A→B ↵ Int List 1→List 2 ↵ Int Mat A→Mat B ↵	OPTN > NUM F2
:fPart *value* **:fPart** *list* **:fPart** *matrix*	**Frac** *value* ↵ **Frac** *list* ↵ **Frac** *matrix* ↵	:fPart A→B :fPart L_1→L_2 :fPart [A]→[B]	Frac A→B ↵ Frac List 1→List 2 ↵ Frac Mat A→Mat B ↵	OPTN > NUM F3
:int *value* **:int** *list* **:int** *matrix*	**Intg** *value* ↵ **Intg** *list* ↵ **Intg** *matrix* ↵	:int A→B :int L_1→L_2 :int [A]→[B]	Intg A→B ↵ Intg List 1→List 2 ↵ Intg Mat A→Mat B ↵	OPTN > NUM F5
:min(*value A, valueB***)**	See example	:Input A, B :Disp min(A, B)	?→A: ?→B ↵ A≤B⇒A◁ B<A⇒B◁	PRGM JUMP F3
:min(*list***)**	**Min(***list***)** ↵	:min(L_1)→X	Min(List 1)→X ↵	OPTN LIST > F1

TI-82 MATH: NUM

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:min(*listA, listB***)** [returns list of smaller of each pair of elements in *listA* and *listB*]	**Min(***listA, listB***)** ↵ [returns the list (*listA* or *listB*) that contains the smallest value]	:min(L_1, L_2)→L_3	Min(List 1, List 2)→List 3 ↵	OPTN LIST > F1
:max(*value A, valueB***)**	See example	:Input A, B :Disp max(A, B)	?→A: ?→B ↵ A≥B⇒A◁ B>A⇒B◁	PRGM JUMP F3
:max(*list***)**	**Max(***list***)** ↵	:For (A, 1, max(L_1)) :[*commands*] :End	For 1→A To Max(List 1) ↵ [*commands*] ↵ Next ↵	OPTN LIST > F2
:max(*listA, listB***)**	**Max(***listA, listB***)** ↵	:max(L_1, L_2)→L_6	Max(List 1, List 2)→List 6 ↵	OPTN LIST > F2

TI-82 MATH: HYP

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:sinh *value* **:sinh** *list*	**sinh** *value* ↵ **sinh** *list* ↵	:Disp sinh .5	sinh .5 ◁	OPTN > HYP F1
:cosh *value* **:cosh** *list*	**cosh** *value* ↵ **cosh** *list* ↵	:Disp cosh .5	cosh .5 ◁	F2
:tanh *value* **:tanh** *list*	**tanh** *value* ↵ **tanh** *list* ↵	:Disp tanh .5	tanh .5 ◁	F3
:sinh^{-1} *value* **:sinh^{-1}** *list*	**sinh^{-1}** *value* ↵ **sinh^{-1}** *list* ↵	:Disp sinh^{-1}{0, 1}	sinh^{-1}{0, 1} ◁	F4
:cosh^{-1} *value* **:cosh^{-1}** *list*	**cosh^{-1}** *value* ↵ **cosh^{-1}** *list* ↵	:Disp cosh^{-1}{2, 3}	cosh^{-1}{2, 3} ◁	F5
:tanh^{-1} *value* **:tanh^{-1}** *list*	**tanh^{-1}** *value* ↵ **tanh^{-1}** *list* ↵	:Disp tanh^{-1}{0, .5}	tanh^{-1}{0, .5} ◁	F6

TI-82 MATH: PRB

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:rand	**Ran#** ↵	:10*rand +1 → R	10 × Ran# +1 → R ↵	OPTN > PROB F4
:*items* **nPr** *number*	*items* **P** *number* ↵	:10 nPr 4 → P	10 **P** 4 → P ↵	OPTN > PROB F2
:*items* **nCr** *number*	*items* **C** *number* ↵	:10 nCr 4 → C	10 **C** 4 → C ↵	OPTN > PROB F3
:*value*! :*list*!	*value*! ↵ *list*! ↵	:10! → A :L_1! → L_2	10! → A ↵ List 1! → List 2 ↵	OPTN > PROB F1
Not available	**P**(*t*) ◁ **Q**(*t*) ◁ **R**(*t*) ◁ **t**(*x*) ◁			OPTN > PROB > F1~F4

TI-82 TEST: TEST / LOGIC

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:= :≠ :>> :≥ :< :≤	= ≠ > ≥ ≤	:A=B→X :If X=0 :Then Disp "NOT EQUAL" :Else Disp "EQUAL" :End	A=B→X↵ If X=0↵ Then "NOT EQUAL"◁ Else "EQUAL"◁ End↵	PRGM > REL F1 ~ F6
:*valueA* **and** *valueB* :*valueA* **or** *valueB* :*valueA* **xor** *valueB* :**not** *value*	*valueA* **And** *valueB* *valueA* **Or** *valueB* **Not** *value*	:"1/(X(X - 2))"→Y_1 :Input X :If X=0 Or X=2 :Then Disp "UNDEFINED" :Else Disp Y_1(X) :End	"1÷(X(X - 2))"→Y1↵ ?→X↵ If X=0 Or X=2↵ Then "UNDEFINED"◁ Else Y1(X)◁ End↵	OPTN >> LOGIC F1 ~ F3

TI-82 MATRX : NAMES / MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
[A] ~ [E] [A](*row, column*)	**Mat** A ~ **Mat** Z **Mat** A[*row, column*]	:[[1,2,3][4,5,6]] → [A] :0 → [A](1,1)	[[1,2,3][4,5,6]] → Mat A ↵ 0 → Mat A[1,1] ↵	OPTN MAT F1
:det *matrix*	**Det** *matrix* ↵	:det[A] → L_1(1)	Det Mat A → List1[1] ↵	OPTN MAT F3
:*matrix*^{T}	**Trn** *matrix* ↵	:[A]^T → [B]	Trn Mat A → Mat B ↵	OPTN MAT F4
:dim *matrix*	**Dim** *matrix* ↵	:dim[A] → L_1	Dim Mat A → List1 ↵	OPTN MAT > F2
:{*row, column*} → **dim** *matrix*	Not available. The dimensions of a matrix may only be defined in Matrix Mode, not within a program.			
:Fill(*value, matrix*)	**Fill**(*value, matrix*) ↵	:Fill(3, [A])	Fill(3, Mat A) ↵	OPTN MAT > F3
:identity *dimension*	**Identity** *dimension* ↵	:identity 5 → [A]	Identity 5 → Mat A ↵	OPTN MAT > F1

TI-82 MATRX : NAMES / MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:randM(*row, column***)** [elements are random integers -9 to 9]	See example	:randM(3,4)→[A]	[In Matrix Mode, define Mat A as 3 × 4] Dim A→List1 ↵ Fix 0 ↵ For 1→R to List1[1] ↵ For 1→C to List1[2] ↵ Ran# × 18-9→Mat A[R,C] ↵ Next ↵ Next ↵	
:augment(*matrixA, matrixB***)**	**Augment(***matrixA, matrixB***)** ↵	:augment([A],[B])→[A]	Augment(Mat A, Mat B) →Mat A ↵	OPTN MAT F5
See example	**Mat→List(***matrix, column***)** ↵	:dim [A]→L_1 :Prompt C :For (R, 1, L_1(1)) :[A](R, C)→L_2(R) :End	"C=" ?→C◁ Mat→List(Mat A, C) → List2 ↵	OPTN MAT F2
:rowSwap(*matrix, rowA, rowB***)**	**Swap** *matrixname, rowA, rowB* ↵ [*matrixname*: A~Z]	:rowSwap ([A], 2, 3)	Swap A, 2, 3 ↵	F4 MAT F1
:row+(*matrix, rowA, rowB***)**	**Row+** *matrixname, rowA, rowB* ↵ [*matrixname*: A~Z]	:row+([A], 1, 4)	Row+ A, 1, 4 ↵	F4 MAT F4

TI-82 MATRX : NAMES / MATH

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:***row**(*value, matrix, row*)	***Row** *value, matrixname, row* ↵ [*matrixname*: A~Z]	:*row(-2, [A], 5)	*Row -2, A, 5 ↵	F4 MAT F2
:***row+**(*value, matrix, rowA, rowB*)	***Row+** *value, matrixname, rowA, rowB* ↵ [*matrixname*: A~Z]	:*row+(3, [A], 1, 2)	*Row+ 3, A, 1, 2 ↵	F4 MAT F3

TI-82 ANGLE

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
angle ° *list* °	*angle* ° *list* °	:Rad :Input A :Disp sin A°	Rad ↵ ?→A ↵ sin A° ◁	OPTN > ANGL F1
degrees ' *minutes* ' *seconds*'	*degrees* ° *minutes* ° *seconds* °	:Input D, M, S :Disp cos D' M' S'	?→D: ?→M: ?→S ↵ cos D ° M ° S ° ◁	OPTN > ANGL F4
angle ^{r} *list* ^{r}	*angle* ^{r} *list* ^{r}	:Input A :Disp A^{r}	?→A ↵ A^{r}◁	OPTN > ANGL F2
Not available	*angle* ^{g} *list* ^{g}			OPTN > ANGL F3
▷ **DMS**	Not available in PRGM mode			
R▷ Pr(*X*, *Y*) [returns r]	**Pol**(*X*, *Y*) [returns r, θ in a list]	:Input X, Y :Disp R▷ Pr(*X*, *Y*)	?→X: ?→Y ↵ Pol(X, Y)→List1 ↵ List1[1] ◁	OPTN > ANGL > F1

TI-82 ANGLE

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
R▷Pθ(*X, Y*) [returns θ]	**Pol**(*X, Y*) [returns r, θ in a list]	:Input X, Y :Disp R▷Pθ(*X, Y*)	?→X: ?→Y↵ Pol(X, Y)→List1↵ List1[2] ◁	Same as above
P▷Rx(*r*, θ) [returns X]	**Rec**(*r*, θ) [returns X, Y in a list]	:Input R, T :Disp P▷Rx(R, T)	?→R: ?→T↵ Rec(R, T)→List1↵ List1[1] ◁	OPTN > ANGL > F2
P▷Ry(*r*, θ) [returns Y]	**Rec**(*r*, θ) [returns X, Y in a list]	:Input R, T :Disp P▷Ry(R, T)	?→R: ?→T↵ Rec(R, T)→List1↵ List1[2] ◁	Same as above

TI-82 DRAW: DRAW

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:ClrDraw	**Cls** ↵	:ClrDraw	Cls ↵	Sketch F1
:Line(*X_1, Y_1, X_2, Y_2***)**	**F-Line** *X_1, Y_1, X_2, Y_2* ↵	:Line(-1, 5, 7, -2)	F-Line -1, 5, 7, -2 ↵	Sketch > LINE F2
:Line(*X_1, Y_1, X_2, Y_2, 0***)** [erases the line]	Not available			
:Horizontal *Y-value*	**Horizontal** *Y-value* ↵	:Horizontal 4.5	Horizontal 4.5 ↵	Sketch > F5
:Vertical *X-value*	**Vertical** *X-value* ↵	:Vertical B	Vertical B ↵	Sketch > F4
:Tangent(*function, X-value***)**	**Tangent** *function, X-value* ↵	:"$2X^2+1$ " → Y_1 :Tangent(Y_1, 3)	"$2X^2+1$" → Y1 ↵ Tangent Y1, 3 ↵	Sketch F2
See example: This program calculates the slope of the tangent line at *X-value* and uses its negative reciprocal to draw the normal line. [Normal lines cannot be drawn at *X-values* for which the derivative is 0 or undefined.]	**Normal** *function, X-value* ↵ [Normal lines cannot be drawn at *X-values* for which the derivative is 0 or undefined.]	:"X(X+3)(X-5)" → Y_1 :DrawF Y_1: Pause :Prompt V :nDeriv(Y_1, X, V) → D :-1/D → M :Y_1(V) - MV → B :"MX + B" → Y_2 :DrawF Y_2	"X(X+3)(X-5)" → Y1 ↵ Graph Y=Y1 ◁ "V=" ? → V ↵ Normal Y1, V ↵	Sketch F3

TI-82 DRAW: DRAW

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:DrawF *f(x)*	**Graph Y=***f(x)* ↵	:"$2X^2+1$"→Y_1 :DrawF Y_1 :Pause	"$2X^2+1$"→Y1 ↵ Graph Y=Y1 ◁	Sketch GRPH F1
Not available in Draw	**Graph r =***r(θ)* ↵			Sketch GRPH F2
Not available in Draw	**Graph (X,Y)=***X(t), Y(t)* ↵			Sketch GRPH F3
See example	**Graph X=***constant* ↵	:Vertical 5	Graph X=5 ↵	Sketch GRPH F4
:Shade(*lowerf, upperf***)** **:Shade(***lowerf, upperf, res***)** [*res*: 1~9]	Incorporated into Inequality Graph commands (see below).	:"$2X^2-6$"→Y_1 :"$-2X^2-6$"→Y_2 :Shade(Y_1, Y_2)	"$2X^2-6$"→Y1 ↵ "$-2X^2-6$"→Y2 ↵ Graph Y>Y1 ↵ Graph Y<Y2 ↵	See below
:Shade(*lowerf, upperf, res, Xleft***)** **:Shade(***lowerf, upperf, res, Xleft, Xright***)** [*res*: 1~9]	Incorporated into Integral Graph command (see below). Otherwise, not available.			

TI-82 DRAW: DRAW

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
See example	**Graph** ∫ *=f(x), Xleft, Xright* ↵ **Graph** ∫ *=f(x), Xleft, Xright, # of divisions* ↵ [#: 1~9]	:"$2X^2-6$"→Y_1 :DrawF Y_1 :Shade(0, Y_1, 1, -5, 5) :Shade(Y_1, 0, 1, -5, 5) :Disp "FN INT" :Disp fnInt(Y_1, X, -5, 5)	"$2X^2-6$"→Y1 ↵ Graph ∫ =Y1, -5, 5, 4 ↵	Sketch GRPH F5
See example	**Graph Y>***f(x)* ↵	:"$2X^2-6$"→Y_1 :DrawF Y_1 :Shade(Y_1, Ymax)	"$2X^2-6$"→Y1 ↵ Graph Y>Y1 ↵	Sketch GRPH > F1
See example	**Graph Y<***f(x)* ↵	:"$2X^2-6$"→Y_1 :DrawF Y_1 :Shade(Ymin, Y_1)	"$2X^2-6$"→Y1 ↵ Graph Y<Y1 ↵	Sketch GRPH > F2
See example	**Graph Y≥***f(x)* ↵	:"$2X^2-6$"→Y_1 :DrawF Y_1 :Shade(Y_1, Ymax)	"$2X^2-6$"→Y1 ↵ Graph Y≥Y1 ↵	Sketch GRPH > F3
See example	**Graph Y≤***f(x)* ↵	:"$2X^2-6$"→Y_1 :DrawF Y_1 :Shade(Ymin, Y_1)	"$2X^2-6$"→Y1 ↵ Graph Y≤Y1 ↵	Sketch GRPH < F4
:DrawInv *f(x)*	**Inverse** *f(x)* ↵	:"X(X+3)(X-5)"→Y_1 :DrawInv Y_1	"X(X+3)(X-5)"→Y1 ↵ Inverse Y1 ↵	Sketch F4

TI-82 DRAW: DRAW

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Circle(*X, Y, radius***)**	**Circle** *X, Y, radius* ↵	:Circle(-2, 5, 7)	Circle -2, 5, 7 ↵	Sketch > F3
:Text(*row, column, "text"***,. .** **.)** **:Text(***row, column, value***,. .** **.)** [*row*: 0~57] [*column*: 0~94]	**Text** *row, column, "text"*,. . . ↵ **Text** *row, column, value*,. . . ↵ [*row*: 1~58] [*column*: 1~127]	:"X^2 " → Y_1 :DrawF Y_1 :Text(10, 5, "Y= X^2 ")	"X^2 " → Y1 ↵ Graph Y=Y1 ↵ Text 10, 8, "Y= X^2 " ↵	Sketch >, > F2

TI-82 DRAW: POINTS

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Pt-On(*X, Y***)**	**PlotOn** *X, Y* ↵	:Pt-On(1, 2)	PlotOn 1, 2 ↵	Sketch > PLOT F2
:Pt-Off(*X, Y***)**	**PlotOff** *X, Y* ↵	:Pt-Off(1, 2)	PlotOff 1, 2 ↵	Sketch > PLOT F3
:Pt-Change(*X, Y***)**	**PlotChg** *X, Y* ↵	:Pt-Change(1, 2)	PlotChg 1, 2 ↵	Sketch > PLOT F4
:Pxl-On(*row, column***)** [*row*: 0~62] [*column*: 0~94]	**PxlOn** *row, column* ↵ [*row*: 1~63] [*column*: 1~127]	:Pxl-On(62, 94)	PxlOn 63, 127 ↵	Sketch >, > PIXL F1
:Pxl-Off(*row, column***)** [*row*: 0~62] [*column*: 0~94]	**PxlOff** *row, column* ↵ [*row*: 1~63] [*column*: 1~127]	:Pxl-Off(62, 94)	PxlOff 63, 127 ↵	Sketch >, > PIXL F2
:Pxl-Change(*row, column***)** [*row*: 0~62] [*column*: 0~94]	**PxlChg** *row, column* ↵ [*row*: 1~63] [*column*: 1~127]	:Pxl-Change(62, 94)	PxlChg 63, 127 ↵	Sketch >, > PIXL F3

TI-82 DRAW: POINTS

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
:Pxl-Test(*row, column***)** [*row*: 0~62] [*column*: 0~94]	**PxlTest** *row, column* ↵ [*row*: 1~63] [*column*: 1~127]	:Pxl-Test(62, 94)	PxlTest 63, 127 ↵	Sketch >, > F4

TI-82 DRAW: STO

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
See VARS: GDB/Picture for info on items stored under each.				
:StorePic Pic*n* [Pic*n*: Pic1~Pic6]	**StoPict** *n* ↵ [*n*: 1~6]	:StorePic Pic1	StoPict 1 ↵	OPTN >, > PICT F1
:RecallPic Pic*n* [Pic*n*: Pic1~Pic6]	**RclPict** *n* ↵ [*n*: 1~6]	:RecallPic Pic2	RclPict 2 ↵	OPTN >, > PICT F2
:StoreGDB GDB*n* [GDB*n*: GDB1~GDB6]	**StoGMEM** *n* ↵ [*n*: 1~6]	:StoreGDB GDB3	StoGMEM 3 ↵	F4 GRPH GMEM F1
:RecallGDB GDB*n* [GDB*n*: GDB1~GDB6]	**RclGMEM** *n* ↵ [*n*: 1~6]	:RecallGDB GDB4	RclGMEM 4 ↵	F4 GRPH GMEM F2

TI-82 KEYBOARD

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
RCL *variable*	Not available			
? [text character]	**?** [text character or Input command]	:Disp "HOW MANY GAMES?" :Input N	"HOW MANY GAMES?" ◁ ?→N↵	PRGM F4
:	:	:0→X: 1→Y	0→X: 1→Y↵	PRGM > F5
"	"	:Disp "TI-82"	"CFX-9850G"◁	ALPHA F2
$L_1 \sim L_6$	**List** *n* [*n*: 1~6]	:Disp L_1	List 1◁	OPTN LIST F1
$L_1 \sim L_6$	**List1 ~ List6**	:Disp L_2	List2◁	F4 STAT LIST F1~F6
U_{n-1}	a_n a_{n+1}	:"U_{n-1} + 2"→U_n	"a_n + 2"→a_{n+1}↵ "a_{n+1} + a_n"→a_{n+2} ↵[not available on TI-82]	VARS > RECR FORM F2~F3
U_{n-1}	a_n a_{n+1}	:"3U_{n-1} - *n*"→U_n	"3a_n - n"→a_{n+1}↵ "a_{n+1} + a_n + n"→a_{n+2} ↵[not available on TI-82]	F4 > RECR n, a_n, ... F2~F3

TI-82 KEYBOARD

TI-82	Casio	Ex: TI-82	Ex: Casio	Location on Casio
V_{n-1}	b_n b_{n+1}	:"V_{n-1} + 2" → V_n	"b_n + 2" → b_{n+1} ↵ "b_{n+1} + b_n" → b_{n+2} ↵ [not available on TI-82]	VARS > RECR FORM F4~F5
V_{n-1}	b_n b_{n+1}	:"3V_{n-1} - n" → V_n	"3b_n - n" → b_{n+1} ↵ "b_{n+1} + b_n + n" → b_{n+2} ↵ [not available on TI-82]	F4 > RECR n, a_n, ... F4~F5
n	n	:"n^2" → V_n	"n^2" → b_n ↵	F4 > RECR n, a_n, ... F1
ABS *value*	**Abs** *value*	:Input V :Disp ABS V	? → V ↵ Abs V◁	OPTN > NUM F1

CASIO Color Menu

Command	Description	Location
P/L-Blue **P/L-Orange** **P/L-Green**	Sets default color for graphs.	SET UP > P/L F1~F3
Blue **Orange** **Green**	Sets color for stat graphs.	F4 STAT COLR F1~F3
BlueG # **OrangeG #** **GreenG #**	Sets color for graphs of functions and recursions.	F4 GRPH COLR F1~F3 [See Prgm Mode Command List for more locations]

CASIO List and Function Memories

Command	Description	Location
ListFile: **File1 ~ File6**	Specifying a File within a program activates the 6 lists stored under it. A total of 36 lists can be stored by the CFX-9850G.	SET UP >, > LIST F1~F6
Function Memory: **f1 ~ f6**	Within a program, RCL (F2) fn (F1~F6) causes the function chosen to be copied to the current program line. Likewise, STO (F1) fn causes the current program line to be stored to the function chosen. While a function may not be stored to function memory using the → command, f1~f6 may be evaluated and sketched.	OPTN >, > F-MEM F1~F3

CASIO Equation Menu

Command	Description	Location
Sim Result	Displays the result of solving the current system of equations (entered in EQN mode) simultaneously.	VARS > EQUA F1
Sim Coeff	Displays a matrix of the coefficients of the current system of equations.	F2
Ply Result	Displays the result of solving the current polynomial (entered in EQN mode) for its roots.	F3
Ply Coeff	Displays a matrix of the coefficients of the current polynomial.	F4

CASIO Symbol Menus

Command	Description	Location
' " ~ * / #	These additional symbols are for use in text display.	SYBL
Engineering Units:		
m	milli (10^{-3})	OPTN >, > E-SYM F1~F5
μ	micro (10^{-6})	
n	nano (10^{-9})	
p	pico (10^{-12})	
f	femto (10^{-15})	
k	kilo (10^{3})	OPTN >, > E-SYM > F1~F5
M	mega (10^{6})	
G	giga (10^{9})	
T	tera (10^{12})	
P	peta (10^{15})	
E	exa (10^{18})	OPTN >, > E-SYM >, > F1

CASIO Complex Numbers Menu

Command	Description	Location
i	The imaginary unit.	OPTN CPLX F1
Abs (*a* + *bi*)	Returns the Absolute Value (r) of the complex number when graphed in the Gaussian plane.	F2
Arg (*a* + *bi*)	Returns the Argument (θ) of the complex number when graphed in the Gaussian plane.	F3
Conj (*a* + *bi*)	Returns the Conjugate of the complex number.	F4
ReP (*a* + *bi*)	Returns the Real Number Part.	F5
ImP (*a* + *bi*)	Returns the Imaginary Part.	F6

R00257887D