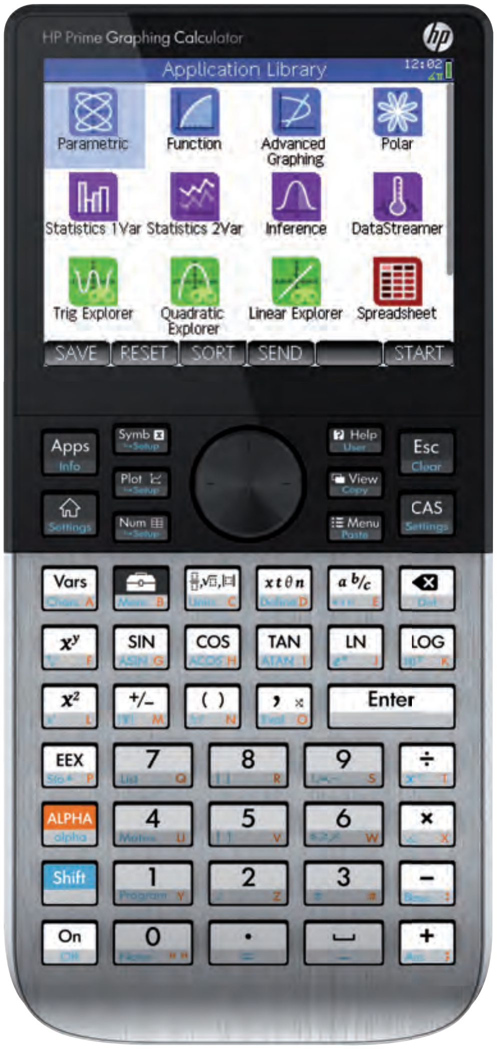
HP Prime Graphing Calculator

The Tutorial Series

This is the first issue of a series of tutorials for the HP Prime, written by Edward Shore. If you have programmed with the HP 39g, 39g or 39gII, you will recognize the programming as the HP Prime programming language (HPPP) is similar. We are using the latest firmware in this series, available on the website.

Visit our dedicated HP Prime portal:
http://www.hp-prime.com



How to start?

1. Press Shift + 1 (Program).

2. Press New. It is the second touch key.

3. Enter the name of the program. Pressing the ALPHA key twice will turn on UPPERCASE ALPHA-LOCK. Pressing ALPHA, Shift, ALPHA will turn on lowercase alpha-lock. To exit any lock, press the ALPHA key one more time. When you're happy with the name, press Enter.

Rules for Program Names:

1. Letters, numbers, and the underscore character (_) only.

2. The program name must start with a letter.

Structure of a HP Prime Program

A HPPP program is encased of an EXPORT - BEGIN - END structure.
The layout is generally like this:

```
EXPORT program_name(arguments)
BEGIN
commands and comments go here
END;
```

Each line containing a command generally must end with a semicolon (;).
A semicolon can by type by pressing ALPHA then the Plus key (+).

Comments can be typed. They are designated by two forward slashes. The slashes are typed by pressing the Divide key (÷). Anything in the line following the two slashes is ignored in running the program.

MORAVIA EDUCATION ©

Preparation and Copyright: MORAVIA Education, a division of MORAVIA Consulting Ltd.
www.moravia-consulting.com
www.hp-prime.com

Date of issue: 04.2014

SQIN

Our first program is SQIN, because "Hello World" programs are so 2000s. SQIN takes a number, squares it, then calculates the reciprocal. In short we are defining a custom function:

SQIN(x) = 1/x^2

Commands:

RETURN: returns a result to the stack (home page). You can return numbers, lists, vectors, matrices, strings, or a combination of these times.
Access: Tmplt, 1. Block, 2. RETURN

```
EXPORT SQIN (X)
BEGIN
RETURN 1/X^2;
END;
```

How to run the programs:

Home Mode - Textbook Entry,
Home Mode - Algebraic Entry,
CAS Mode:

Type the program name. Follow the name with parenthesis and enclose the required arguments.

Or use the Toolbox (top row of white keys, 2nd key from the left, it looks like a tool box), select the User touch key, select the program, and input the required arguments.

Home Mode - RPN Entry:

Enter each argument, separate each entry by pressing the Enter key. Type the name, and in the parenthesis state the number of arguments.

For example, if the program TEST has four arguments, the RPN stack would like this:

4: argument_1
3: argument_2
2: argument_3
1: argument_4
TEST(4) to run the program.

Examples to try with SQIN:

SQIN(5) returns .04
SQIN(36) returns .000771604938

TIP!

You can check the syntax of the program just by pressing the Check soft key in the program editor. HP Prime will Inform you if there is a syntax error and attempt to point you to the error. If there are no syntax errors, the Prime states "No errors in the program". I use the Check Command all the time.

MOPMT

LOCAL: Declares any variables to be local to the program. In other words, the variables are created, used, possibly displayed during program execution, and deleted at program termination.
Access: Tmplt, 4. Variable, 1. LOCAL

MOPMT calculates the monthly payment of a loan. The arguments are: the loan amount (L), the interest rate (R), and the number of months (M).

```
EXPORT MOPMT(L,R,M)
BEGIN
LOCAL K:=R/1200;
K:=L*K/(1-(1+K)^-M);
RETURN "Payment ="+K;
END;
```

TIP!

You can declare local variables and assign an initial value at the same time. For example: LOCAL K:=1; stores 1 in K and makes K a local variable.

Examples:
MOPMT(4000, 9.5, 30) returns 150.317437565
MOPMT(370000, 3.5, 360) returns 1661.46534383

Try this and next time in the series I will highlight other things we can do with HPPP.

TIP!

Use RETURN, TEXTOUT_P, and PRINT to return custom strings, which combine results, messages, and calculations. Parts are connected with a plus sign.

HP Prime Graphing Calculator

The Tutorial Series

This is the second issue of a series of tutorials for the HP Prime, written by Edward Shore. In this session, we will cover MSGBOX, IF-THEN-ELSE, PRINT, and the FOR loop. If you have programmed with the HP 39g, 39g or 39gII, you will recognize the programming as the HP Prime programming language (HPPP) is similar. We are using the latest firmware in this series, available on the website.

Visit our dedicated HP Prime portal:
http://www.hp-prime.com

How to start?

1. Press Shift + 1 (Program).

2. Press New. It is the second touch key.

3. Enter the name of the program. Pressing the ALPHA key twice will turn on UPPERCASE A LPHA-LOCK. Pressing ALPHA, Shift, ALPHA will turn on lowercase alpha-lock. To exit any lock, press the ALPHA key one more time. When you're happy with the name, press Enter.

Rules for Program Names:

1. Letters, numbers, and the underscore character (_) only.

2. The program name must start with a letter.

Structure of a HP Prime Program

A HPPP program is encased of an EXPORT - BEGIN - END structure. The layout is generally like this:

```
EXPORT program_name(arguments)
BEGIN
commands and comments go here
END;
```

Each line containing a command generally must end with a semicolon (;). A semicolon can by type by pressing ALPHA then the Plus key (+).

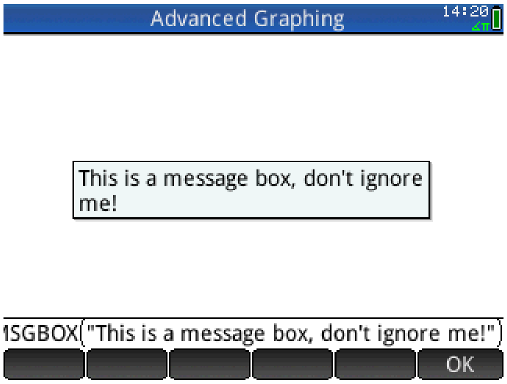
Comments can be typed. They are designated by two forward slashes. The slashes are typed by pressing the Divide key (÷). Anything in the line following the two slashes is ignored in running the program.

MSGBOX

MSGBOX takes a string a makes a pop-up message box. Program execution stops until you press a key to acknowledge the message.

Access: Cmds, 6. I/O, 8. MSGBOX

The program COMLOCK: Imagine that you are in charge of setting the combinations for the good, old-school combination locks. This program gives three digit combinations through the use of MSGBOX.

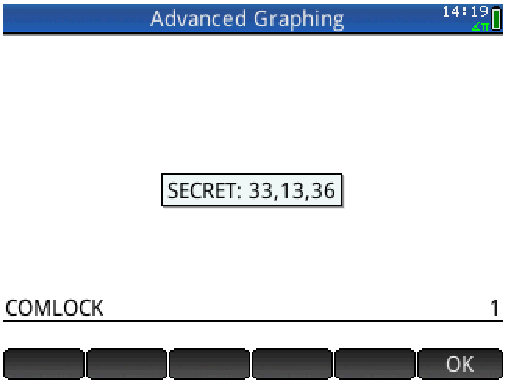


This is a message box, don't ignore me!

ISGBOX("This is a message box, don't ignore me!")

```
EXPORT COMLOCK()
BEGIN
LOCAL L0;
L0:=RANDINT(3,0,39);**
MSGBOX("SECRET: "+L0(1)+","+L0(2)+","+L0(3));
END;
```

Here is a sample output for COMLOCK:



SECRET: 33,13,36

COMLOCK 1

Other commands that are featured:

RANDINT(n, a, b) generates a list of n integers between a and b. You can leave n out if you desire a single random integer. Picks may be repeated.

The HP Prime's default list variables are designated L0 through L9.

Tip!

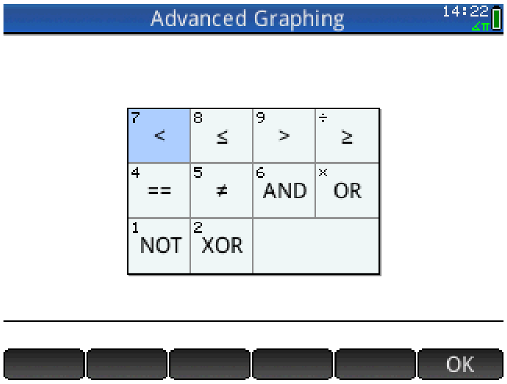
You can leave out the ELSE part if you only want to test to see if a condition is true. Access the simple IF-THEN structure by pressing Tmplt, 2. Branch, 1. IF THEN.

IF-THEN-ELSE

IF-THEN-ELSE: Program structure:
IF condition THEN
do if the condition is true;
ELSE
do if the condition is false;
END;

Access: Tmplt, 2. Branch, 2. IF THEN ELSE

Access <, ≤, ==, etc. by pressing Shift, 6. Note that the double equals is needed to check equality.



PRINT

The PRINT command prints a sting, result, or a combination of both onto the Prime's Terminal screen. If PRINT is used, the program will end on the terminal (text output) screen. Press a button to exit.

You can access the terminal screen at any time by pressing the ON button, holding it, and then pressing the Divide (÷) button.

Access: Cmds, 6. I/O, 9. PRINT

Tip!

To clear the terminal screen, type PRINT(). This is a good way to clear the terminal screen and I usually use this at the beginning of any program if PRINT is going to be used later on.

The program QROOTS (yet one more quadratic solver, sorry for not being original guys and gals), demonstrates the use of IF-THEN-ELSE and PRINT.

Here I set the setting variable HComplex to 1, which allows for complex number results.

```
EXPORT QROOTS(A,B,C)

BEGIN
LOCAL D;
PRINT();
HComplex:=1;
D:=B^2-4*A*C;
IF D≥0 THEN
PRINT("Roots are real.");
ELSE
PRINT("Roots are complex.");
END;
```

```
PRINT((-B+√D)/(2*A));
PRINT((-B-√D)/(2*A));
END;
```

Examples:

QROOTS(1,5,8) returns:

Roots are complex.

-2.5+1.32287565553*i
-2.5-1.32287565553*i

QROOTS(2,-4,-8) returns:

Roots are real.

3.2360679775
-1.2360679775

FOR

This section will explore the basic **FOR** structure:

```
FOR variable FROM start TO end DO
commands;
END;
```

All the commands in the loop will be executed a set number of times. Each time a loop finishes, the *variable* increases by one. The loop terminates when *variable=end*.

Access: Tmplt, 3. LOOP, 1. FOR

The program SUMDIV takes any integer and adds up the sum of its divisors. For example, the divisors of 12 are 1, 12, 2, 3, 4, and 6. The sum is 28.

Featured Commands in SUMDIV:

idivis: idivis(*integer*) returns a sequence of all of the divisors if *integer*. Access: Toolbox, CAS, 5. Integer, 1. Divisors

Any CAS command used in programming will be preceded by "CAS." Not all CAS commands can be used in HP Prime programming at this time.

DIM: returns the dimensions of a sequence, string, or matrix. DIM must be used instead of SIZE to prevent a Bad Argument error.

For sequences or vectors, DIM returns the length in a list {length}.
For strings, DIM returns length as a number.
For matrices, DIM returns the list {number of rows, number of columns}.

Access: Cmds, 1. Strings, 9. DIM

The program:

```
EXPORT SUMDIV(N)
BEGIN
LOCAL S:=0,K,mdiv,ldiv;
mdiv:=CAS.idivis(N);
ldiv:=DIM(mdiv);
FOR K FROM 1 TO ldiv(1) DO
S:=S+mdiv(K);
END;
RETURN S;
END;
```

Examples:
SUMDIV(12) returns 28.
SUMDIV(24) returns 60.
SUMDIV(85) returns 108.

HP Prime Graphing Calculator

The Tutorial Series

This is the third issue of a series of tutorials for the HP Prime, written by Edward Shore. This tutorial is going to cover a lot, each with some new programming commands. I hope you are ready for the intensity! If you have programmed with the HP 39g, 39g or 39gII, you will recognize the programming as the HP Prime programming language (HPPP) is similar. We are using the latest firmware in this series, available on the website.

Visit our dedicated HP Prime portal:
http://www.hp-prime.com

How to start?

1. Press Shift + 1 (Program).

2. Press New. It is the second touch key.

3. Enter the name of the program. Pressing the ALPHA key twice will turn on UPPERCASE ALPHA-LOCK. Pressing ALPHA, Shift, ALPHA will turn on lowercase alpha-lock. To exit any lock, press the ALPHA key one more time. When you're happy with the name, press Enter.

Rules for Program Names:

1. Letters, numbers, and the underscore character (_) only.

2. The program name must start with a letter.

Structure of a HP Prime Program

A HPPP program is encased of an EXPORT - BEGIN - END structure. The layout is generally like this:

```
EXPORT program_name(arguments)
BEGIN
commands and comments go here
END;
```

Each line containing a command generally must end with a semicolon (;). A semicolon can by type by pressing ALPHA then the Plus key (+).

Comments can be typed. They are designated by two forward slashes. The slashes are typed by pressing the Divide key (÷). Anything in the line following the two slashes is ignored in running the program.

WHILE, INPUT, KILL

HP Prime Program: TARGET. TARGET is a game where you provide a guess to get a desired number. If you miss, the calculator will tell you if number is higher and lower. At the end of the game, the calculator gives you how may picks you needed to get the target number.

WHILE: Repeat a number of commands while a specific condition is test.

```
WHILE condition is true DO
commands
END;
```

Access: Tmplt, 3. Loop, 5. WHILE

Caution: Watch your ENDs! Make sure an END is with each loop and the program itself. Press the soft key Check to check your work.

INPUT: Creates an input screen for variables. On the HP Prime, the input can asked for more than one input. TARGET demonstrates INPUT with one prompt.

One Variable:

```
INPUT(variable, "title", "label", "help text")
```

Multi-Variable:

```
INPUT(list of variables, "title", list of "labels", list
of "help text")
```

Note: Pressing Cancel will store a 0 in variable. You may include code of what to do if the user presses Cancel, but it is not required.
Access: Cmds, 6. I/O, 5. INPUT

KILL: Terminates program execution. Nothing dies, I promise.

Access: Tmplt. 1. Block, 3. KILL

Program:

```
EXPORT TARGET()
BEGIN
LOCAL C:=0, N:=RANDINT(1,20), G:=-1;
WHILE G≠N DO
C:=C+1;
INPUT(G,"Guess?","GUESS:","1 - 20");

IF G==0 THEN
KILL;
END;

IF G < N THEN
MSGBOX("Higher");
END;

IF G > N THEN
MSGBOX("Lower");
END;

END;

MSGBOX("Correct! Score: "+C);

END;
```

Try it and of course, you can adjust the higher limit. Here is something for you to try with TARGET:

1. Add a limited amount of guesses.
2. Can you display the list of guesses?

WHILE, INPUT, KILL

HP Prime Program: TARGET. TARGET is a game where you provide a guess to get a desired number. If you miss, the calculator will tell you if number is higher and lower. At the end of the game, the calculator gives you how may picks you needed to get the target number.

ULAM Algorithm: take an integer n. If n is even, divide it by 2. If n is odd, multiply it by 3 and add 1. ULAM counts how many steps it takes to get n to 1.

REPEAT:
Access: Tmplt, 3. Loop, 6. REPEAT

Featured:
CONCAT(list1, list2): Melds list1 and list2 into one.

Access: Toolbox, Math, 6. List, 4. Concatenate

```
EXPORT ULAM(N)
BEGIN
LOCAL C:=1, L0:={N};

REPEAT

IF FP(N/2)==0 THEN
N:=N/2;
ELSE
N:=3*N+1;
END;

C:=C+1;
L0:=CONCAT(L0,{N});
UNTIL N==1;

MSGBOX("NO. OF STEPS="+C);
RETURN L0;
END;
```

Examples:

```
ULAM(5) returns:
Message Box: "NO. OF STEPS=6"
List: {5, 16, 8, 4, 2, 1}

ULAM(22) returns:
Message Box: "NO. OF STEPS=16"
List: {22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8,
4, 2, 1}
```

GETKEY

The next section will introduce a super-important command, GETKEY. We will be working with GETKEY over the entire series.

The Program KEYNO: The person presses key presses. Which each key press, the code returns to the terminal screen. The program terminates when the Enter key is pressed.

GETKEY: Returns the key code of last key pressed. The Prime's key map is below. (Picture is from the HP Prime User's Guide)

Access: Cmds, 6. I/O, 4. GETKEY

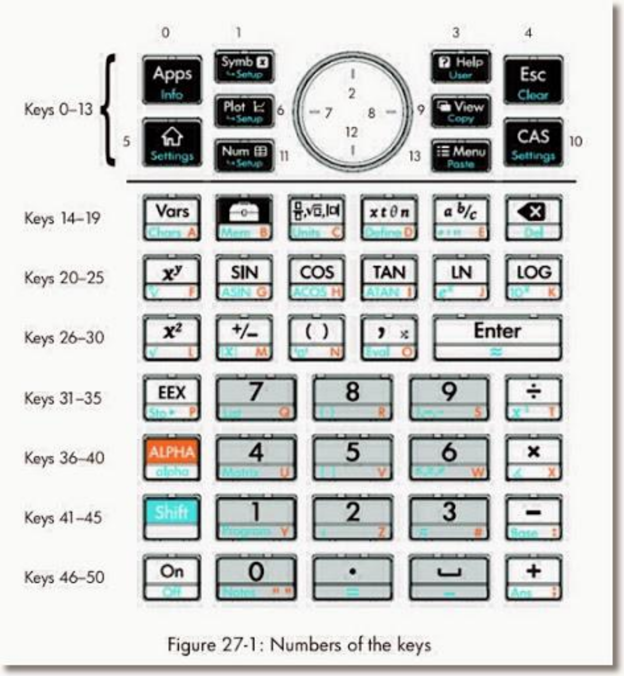


Figure 27-1: Numbers of the keys

```
EXPORT KEYNO()
BEGIN
LOCAL K;
PRINT();
PRINT("Press any key to get its code.");
PRINT("Press Enter to exit.");

REPEAT
K:=GETKEY;
IF K ≥ 0 THEN
PRINT(K);
END;
UNTIL K==30;

END;
Example Key Codes:
33: 8 key
2: up
7: left
8: right
12: down
50: plus
45: minus
```

HP Prime Graphing Calculator

The Tutorial Series

This is the fourth issue of a series of tutorials for the HP Prime, written by Edward Shore. This session will cover CHOOSE and CASE. If you have programmed with the HP 39g, 39g or 39gII, you will recognize the programming as the HP Prime programming language (HPPP) is similar. We are using the latest firmware in this series, available on the website.

Visit our dedicated HP Prime portal:
http://www.hp-prime.com

How to start?

1. Press Shift + 1 (Program).

2. Press New. It is the second touch key.

3. Enter the name of the program. Pressing the ALPHA key twice will turn on UPPERCASE A LPHA-LOCK. Pressing ALPHA, Shift, ALPHA will turn on lowercase alpha-lock. To exit any lock, press the ALPHA key one more time. When you're happy with the name, press Enter.

Rules for Program Names:

1. Letters, numbers, and the underscore character (_) only.

2. The program name must start with a letter.

Structure of a HP Prime Program

A HPPP program is encased of an EXPORT - BEGIN - END structure. The layout is generally like this:

```
EXPORT program_name(arguments)
BEGIN
commands and comments go here
END;
```

Each line containing a command generally must end with a semicolon (;). A semicolon can by type by pressing ALPHA then the Plus key (+).

Comments can be typed. They are designated by two forward slashes. The slashes are typed by pressing the Divide key (÷). Anything in the line following the two slashes is ignored in running the program.

CHOOSE and CASE

CHOOSE: Creates a pop up choose box, similar to what you see when you click on a soft menu. There are two syntaxes for CHOOSE:

Simple Syntax (up to 14 options):
CHOOSE(var, "title string", "item 1", "item 2", ... , "item n");

List syntax (infinite amount of items):
CHOOSE(var, "title string", {"item 1", "item 2"});

Choosing item 1 assigns the value of 1 to var, choosing item 2 assigns the value of 2 to var.

Access: Cmds, 6. I/O, 1. CHOOSE

CASE: Allows for different test cases for one variable. Also includes a default scenario (optional).

```
CASE
IF test 1 THEN do if true END;
IF test 2 THEN do if true END;
...
DEFAULT commands END;

Access: Cmds, 2. Branch, 3. CASE
```

Let's look at two programs to demonstrate both CHOOSE and CASE.

TERMVEL - Terminal Velocity of an Object

```
EXPORT TERMVEL()
BEGIN

LOCAL L0:={9.80665,32.174},
L1:={1.225,.0765},
L2:={.47,1.05,1.15,.04},C,K,M,A,T;

CHOOSE(C,"Units","SI","English");

CHOOSE(K,"Type of Object","Sphere","Cube",
"Cylinder","Tear-Shaped");

INPUT({M,A},"Object",
{"M=","A="},{"Mass","Surface Area"});

T:=√((2*M*L0(C))/(L1(C)*A*L2(K)));

MSGBOX("Terminal Velocity="+T);
RETURN T;
END;
```

Examples:

Sphere, SI Units, M = .05 kg, A = .0028 m^2
Terminal Velocity: T = 24.6640475387 m/s

Cube, US Units, M = 1.2 lb, A = .3403 ft^2
Terminal Velocity: T = 53.149821209 ft/s

TIP: Use the IF THEN ELSE structure with INPUT to execute a set of default instructions if the user presses cancel. INPUT returns a value of 0 if ESC or cancel is pressed, and 1 if a value is entered.

```
IF INPUT(...) THEN
commands if values are entered
ELSE
commands if Cancel is pressed
END;
```

Default values can be assigned to values as an optional fifth argument for INPUT.

```
INPUT(var, "Title", "Prompt",
"Help", default value)
```

The type of variable maybe set to other than real numbers. Just remember to store such type before the INPUT command. For example, if you want var to be a string, store an empty string:

```
var:=" ";
```

AREAC - Area of Circles, Rings, and Sectors

```
EXPORT AREAC()
BEGIN
LOCAL C,R,S,θ,A;
CHOOSE(C,"Areas","1. Circle","2. Ring","3. Sector");

INPUT(R, "Input Radius", "R =");

CASE
IF C==1 THEN A:=π*R^2; END;

IF C==2 THEN
INPUT(S,"Small Radius","r=");
A:=π*(R^2-S^2);
END;

IF C==3
INPUT(θ, "Angle", "θ=");
\\ Assume you are in the correct angle mode
IF HAngle==1 THEN
\\ Test Angle Mode
θ:=θ*π/180;
END;
A:=θ*R^2/2;
END;

END;

MSGBOX("Area is "+A);
RETURN A;
END;
```

Examples:

R = 2.5, r = 1.5, θ = π/4 radians or 45°

Circle: 19.6349540849
Ring: 12.5663706144
Sector: 2.45436926062