

«

"TITLE"

PROGRAMMING FOR BEGINNERS

{

{ "option 1" * program 1 * }
{ "option 2" * program 2 * }

{ "option 3" * program 3 * }

By Marcos Navarro

}

1 CHOOSE

« EVAL » IFT

»

«

18 IF > THEN 1000 0.5 BEEP

ELSE 900 0.5 BEEP

END

HP-50G

»

Graphing Calculator

PROGRAMMING FOR BEGINNERS

User RPL Language

HP-50G Graphing Calculator

First Edition

© 2025 Marcos Navarro

While every effort has been made to ensure the accuracy of the information presented in this book, the author makes no warranties, expressed or implied, and disclaims any liability for errors or omissions. The reader assumes full responsibility for the use of this information.

All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the author.

ISBN: 9798311579292

CONTENT

INTRODUCTION



1. Let's Get Started!	1
2. Variables and Subroutines.	19
3. Strings and Lists.	31
4. Conditional Structures.	45
5. Data Input and Output (Part I).	59
6. Loops (Part I).	67
7. Vectors and Matrices.	77
8. Logic Operations	85
9. Loops (Part II).	95
10. Data Input and Output (Part II).	101
11. Advanced Graphics Handling.	111
12. 2D Graphics.	123
13. Miscellaneous.	135

The Chapter 3 of the HP-50G *“Advanced User’s Reference Manual”* or *“AUR Manual”*, details all the calculator’s commands and functions.

Use the *Advanced User’s Reference Manual* to learn about the following commands:

0 OFF	5 SQ	10 WITHOUT
1 MEM	6 FP	11 STO
2 DUP	7 +	12 VARS
3 SWAP	8 ABS	13 PGDIR
4 DROP	9 SDR	14 TRN

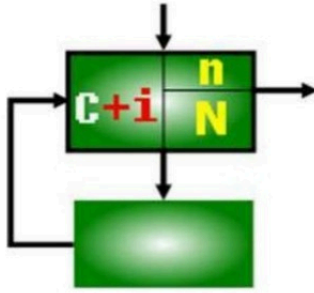
1.4 Let's write our first programs.

Take your HP 50G, press the key  and then the key .

The symbols « and » must appear. These are the program delimiters. All commands and other objects of our *User RPL* program must be written within this symbols.



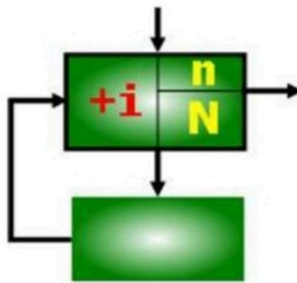
figure 1.5



Flowchart for a FOR...STEP...END structure

Similarly we can replace START...NEXT with START STEP:

« *start end* START...*increment* STEP »



Flowchart for a START...STEP...END structure

Decreases

What would the previous program look like if we wanted the frequency to be reduced from 1000 to 100?

One solution would be:

«
100 1000 FOR FREQ

OR: Takes two logical values from the stack and if any of them is true, it results in a 1. In other words, the OR operation will only be false when both given values are false.

A	B	A B OR
0	0	1
0	1	1
1	0	1
1	1	0

The OR operation is also known as logical sum.
 $A+B$

XOR: Takes two logical values from the stack and if they are equal, the result is 0. In other words, the XOR operation will only be true when the two given values are different.

A	B	A B XOR
0	0	0
0	1	1
1	0	1
1	1	0

The XOR operation is actually a combination of other operations.
 $A \oplus B$

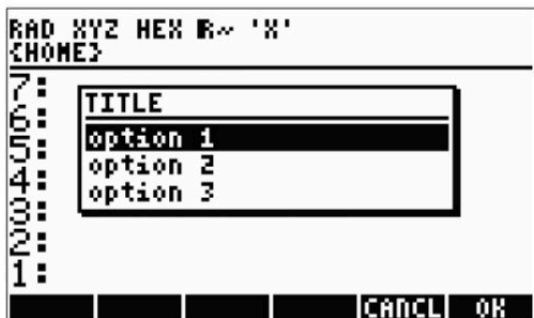
10.1 The CHOOSE menu.

This menu provides an elegant way to ask users to choose between several options.

Let's see an example:

```
« "TITLE"  
{ { "option 1"« program1 » }  
{ "option 2"« program2 » }  
{ "option 3"« program3 » }  
} 1 CHOOSE »
```

The created menu is shown below:



When an option is selected, its corresponding program is pushed onto the stack unevaluated, and a '1' is also pushed to the stack to indicate that an option has been selected.

If the ON key is pressed, and this is done when you want to exit the menu without choosing any option, a '0' will simply appear on the stack.

The HP-50G provides a set of commands for graphical representations. This section will show how to create programs to graph functions with cartesian coordinates.

12.1 Functions.

The first thing to do is to select the type of graphical representation with the **FUNCTION** command. The **DRAW** command is used to create graphical representations. This command requires the **EQ** variable that will contain the function to be plotted.

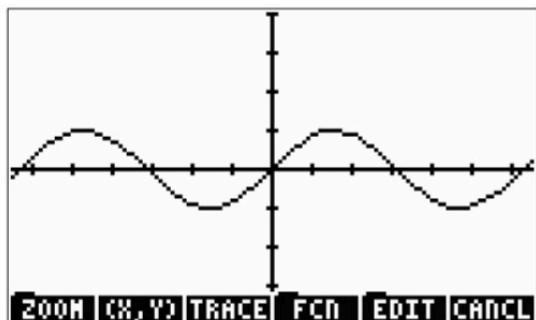
STEQ command, which takes the function from the stack and stores it in **EQ**.

DRAX command draws the Cartesian axes.

E.g.:

Let's write a program to draw the graph of the function $y=\sin(x)$. The code is:

```
« {PICT PPAR} PURGE  
'Y=SIN(X)' STEQ RAD DRAX DRAW  
PICTURE »
```



Challenges:

(a) Write a program that displays time like an analog clock:



(b) Write a program that displays the calendar for the month and updates the day and time each time it is run:

February 2025						
MO	TU	WE	TH	FR	SA	SU
					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		

(c) Write a program that measures the time taken by another program to run.

Unleash the power of your HP 50g with this practical programming guide for beginners. With clear examples and practical exercises, this book will teach you how to program your calculator quickly and easily.

Written with a fundamentally didactic approach, each chapter guides you step by step on a journey to explore the exciting world of User RPL Language.

This book is intended for those who are new to HP-50G calculator programming.

FIND THIS BOOK ON:

- Amazon.com**
- Google Books**
- Google Play Books**

