Binary Numbers

Teachers Teaching with Technology™
Professional Development from Texas Instruments

Answers

7 8 **9** **10** 11 12

TI-Nspire | Coding | Student | 60 min

Binary

Decimal is the most common counting system for humans consisting of the ten familiar digits: 0, 1, 2 … 9. It is certainly no coincidence that human hands generally consist of 10 digits. The most common counting system for computers consists of just two digits: 0, 1 aligning appropriately to the most common electrical states; on and off. A single binary digit is referred to as a bit, a four bit number is referred to as a nibble and an eight bit number is referred to as a byte. This last term is most familiar when dealing with very large quantities such as kilobyte, megabytes or gigabytes[1]. In this activity you will learn how to:

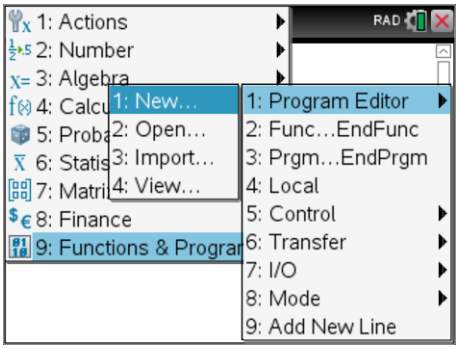
- Write a simple program to convert a decimal number to binary
- Convert decimal number to binary (by – hand)
- Work with ASCII codes

Coding the Algorithm

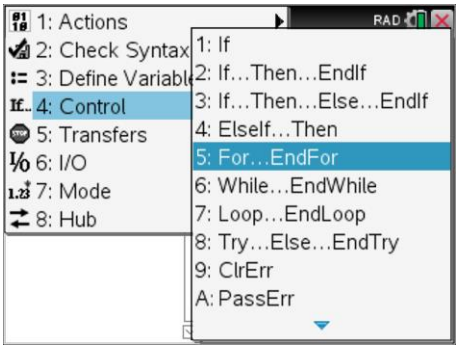
Instructions:

Start a new document; insert a calculator application followed by a program.

Call the program: Binary



The program will start with a 'loop' referred to as a **For** loop. The **For** loop executes a set of commands a specified number of times, it will be used to count from one number to a higher number with the output being displayed as a binary number.



From the menu select: **Control** > **For … EndFor**

Complete the **For** command: **For n, 32, 64**

The first time through the loop 'n' will equal 32, the second time it will equal 33 and so it will continue until n = 64.

Note that when the **For** loop is selected, the EndFor command is automatically placed into the program, effectively working as a set of parenthesis bracketing the commands that will be executed.

[1] Kilobyte = 1000 bytes, Megabyte = 1,000,000 bytes, Gigabyte = 1,000,000,000 bytes.

Author: P. Fox

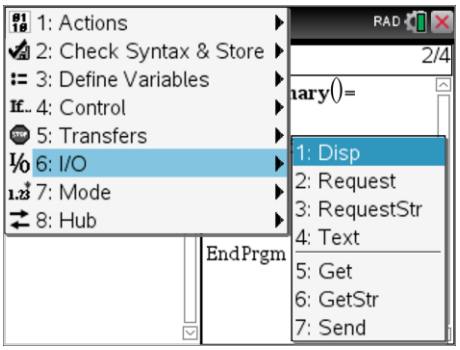
Press **ENTER** to move to the next line, then from the menu select:

I/O > **Disp**

This line will be used to display the 'binary' number.

Immediately following the Disp command type: **n**

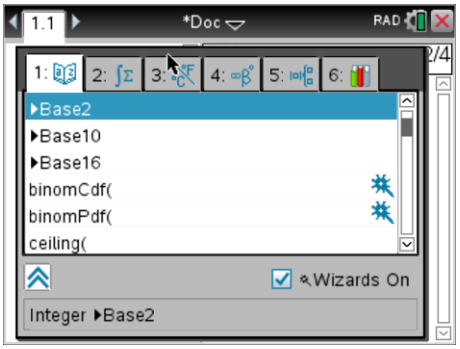
Don't press ENTER just yet, the value of n needs to be converted to a binary number.



Use the Catalogue key to locate and select the Binary number conversion command. Binary numbers are also referred to as 'base 2'.

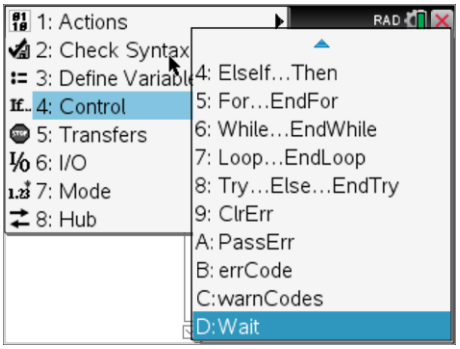
The complete display command should appear as:

Disp n ▸Base2



The program is ready to run, the only problem is that the numbers would go past two quickly to follow. A **Wait** command can be used to slow things down!

Scroll to the bottom of the **Control** menu to locate the **Wait** command then include an appropriate wait time. Start with 1 second, it can be increased or decreased later.

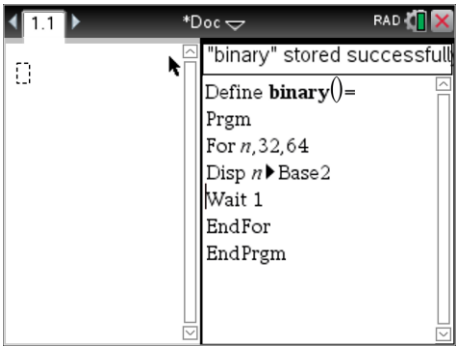


The program is ready to be saved. Press CTRL + B (these keys may be pressed sequentially).

If everything has been entered correctly, the top of the programming application will display the message:

"binary" stored successfully

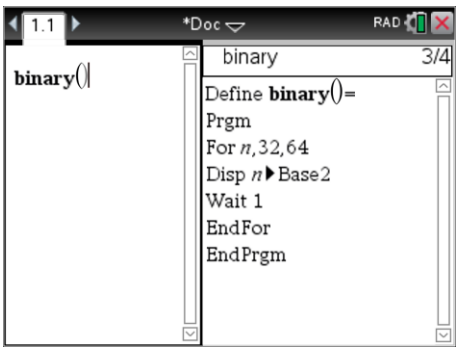
Congratulations, you program is complete and ready to run!



To shift focus from the programming window to the calculator window press **CTRL** + **TAB**, similar to **ALT** + **TAB** in Windows™.

While in the calculator application the program can be run by pressing the **VAR** key and selecting the program Binary, press **ENTER** to run the program.

Note: Binary numbers are written with 0B at the start (left) to signify that they are binary rather than decimal.



Author: P.Fox

Question: 1.

Watch the numbers carefully and observe the pattern or sequence as the binary numbers count from 32 to 64. Look carefully at the last number: 64.
Based on your observations, write down the binary numbers for:

a) 65 0B1000001
b) 66 0B1000010
c) 67 0B1000011
d) 68 0B1000100
e) 69 0B1000101
f) 70 0B1000110

Note: You can edit the program to check your answers. Change the values accordingly in the **For** command, press CTRL + B to save the changes and then run the program again.

Understanding Binary

The two states for binary numbers 0 (off) and 1 (on) make it easy to count in binary. When counting in base 10, each subsequent place holder represents a power of 10. The first column represents units (10^0); the second column is 'tens' (10^1); the third column is 'hundreds' (10^2) and so on. For base 2 each subsequent place holder represents a power of 2. The first column is 2^0; the second column is 2^1; the third is 2^2; and so on. To see how a number such as 79 is written, consider all the powers of 2 that could be combined to form this number. $2^7 = 128$, so 2^7 is too big. $2^6 = 64$, whilst not quite big enough, smaller powers of 2 can be added until 78 is reached. $2^5 = 32$, since $64 + 32 > 78$, then including 2^5 would produce a number at least as big as 96, too big. Similarly including 2^4 would make the result too big since $64 + 16 > 78$ but including $2^3 = 8$ is okay since $64 + 8 = 72$, more powers of 2 are required. The table below shows how 78 can be written by adding up selected powers of 2.

Place holders	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
79	0	1	0	0	1	1	1	1

$79 = 0 \times 128 + 1 \times \mathbf{64} + 0 \times 32 + 0 \times 16 + 1 \times \mathbf{8} + 1 \times \mathbf{4} + 1 \times \mathbf{2} + 1 \times \mathbf{1}$

Question: 2.

Write the first 16 numbers in Binary. Adjust your program to check your answers.
0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000

What is ASCII?

The **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange was established in the 1960's. The code represents an internationally recognised means of transmitting text information via numbers. The system allows for upper and lower case letters, symbols and unprinted keyboard controls to be transmitted using binary numbers, the native language for electronic devices. Characters are sent as a 'byte'. As the ASCII character set consists of 128 characters, this always leaves the 'most significant bit' unassigned. In some systems a parity[2] check is used for this unassigned bit. For simplicity, the unassigned bit will be left as a zero and only upper and lower case letters will be explored.

[2] Parity Check: An odd parity check means that 0 or 1 is assigned to the most significant bit so the quantity of 1's in the byte is odd.
Example 1: "d" = 100(decimal) or 0110 0100 (binary). There are three 1's in this binary number so the most significant bit (left) is zero so there remains an odd quantity of 1's.
Example 2: "Y" = 89 (decimal) or 0101 1001 (binary). There are four 1's in this binary number so the most significant bit would be changed to a 1 so an odd quantity of '1's would be in the number: 1101 1001

Author: P.Fox

ASCII Table – (Letters only)

A = 65	B = 66	C = 67	D = 68	E = 69	F = 70	G = 71	H = 72	I = 73	J = 74
K = 75	L = 76	M = 77	N = 78	O = 79	P = 80	Q = 81	R = 82	S = 83	T = 84
U = 85	V = 86	W = 87	X = 88	Y = 89	Z = 90				
a = 97	b = 98	c = 99	d =100	e = 101	f = 102	g = 103	h = 104	i = 105	j = 106
k = 107	l = 108	m = 109	n =110	o = 111	p = 112	q = 113	r = 114	s = 115	t = 116
u = 117	v = 118	w = 119	x = 120	y = 121	z = 122				

To transmit the word: "fox" (all lower case) would involve sending the numbers: 102, 111 and 120. As a binary transmission with no parity check, this would be:

0110 0110 0110 1111 0111 1000

This transmission could easily be changed to all uppercase as FOX:

0100 0110 0100 1111 0101 1000

Question: 3.

Use the ASCII table to convert your name (all lowercase) into decimal numbers and the corresponding binary representation. (No parity check). Answers will vary

Question: 4.

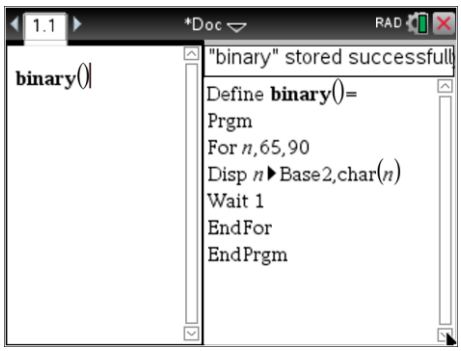
Rewrite your name (all uppercase) in Binary.

Hint: The difference between uppercase and lower case ASCII characters is 32. Answers will vary, however the only difference in each binary number will be the 6th digit from the right. (LSB)

Coding Questions

Question: 5.

Change the **For** loop to start at 65 and finish at 90.
Immediately after the "Disp n▸Base2" command, place a comma and **char**(**n**) Save the program (Ctrl + B) and then run it. What does char(n) do?
Displays the corresponding ASCII character.

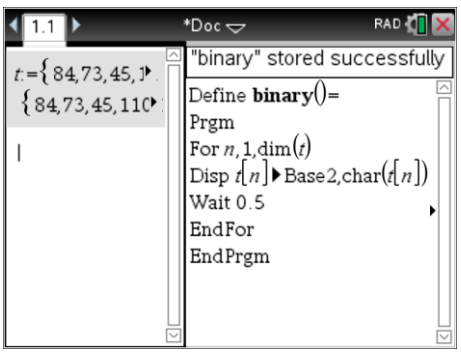


Question: 6.

Use the calculator to store the following numbers in a list called: 't'.
t:={84, 73, 45, 110, 115, 112, 105, 114, 101}

Change the program to match the one shown opposite. Note that the **For** loop starts at 1 and ends at Dim(t), which refers to the dimension, number of entries, in list T.
t[n] calls up element 'n' in list T.

Save and run this version of the program. What information is displayed on the screen? The letters of TI-nspire with the binary number for each character is displayed.



Author: P.Fox

Teacher Notes / Comments

This activity is designed to introduce students to the notion of binary numbers which can then be used to send information via light signals using the TI-Innovator. The ultimate goal is to have students understand digital communications and ultimately how to send data using light, the basis for the National Broadband Network.

Teachers are encouraged to have students explore other properties of binary numbers such as "What do all the binary numbers that end in '1' have in common?"

There are many programming challenges that can be provided to students based on this activity, however it would be appropriate to have students complete the "10 Minutes of Coding" exercises first. A sample challenge is to have students write their own program to convert a decimal number into a binary number or binary 'list'.

Example:

```
Request "Number",n
p:=int(log(n,2))
b:={}
For m,0,p
    If n – 2^(p – m) ≥ 0 Then
        b:=augment(b,{1})
        n:=n – 2^(p – m)
        Else
        b:= augment(b,{0})
    EndIf
Endfor
Disp b
```

Other simpler programs may exist, however this one leaves each 'digit' of the binary number in a list that can be used later for some TI-Innovator projects.

Author: P.Fox