

Counting Ones

Cordell Bonnieux

Blueprint:

Inputs: A user inputted integer, I will call it decimal.

Outputs: A Prompt asking for input from the user, and a prompt which displays the number of '1's in the binary representation of decimal.

Calculations: I will need to create a for loop which repeatedly divides decimal until it reaches 0. On each iteration I need to store the remainder in an array which I will call binary.

Variables: decimal(integer), binary(integer array), remainder(integer), counter(integer), i(integer)

Constants: n/a – none needed

Pseudocode:

Start

Print "Enter an integer between 0 and 99."

Read int to **decimal**

If **decimal** < 0 *or* **decimal** > 99

Restart Program (or method/function)

Print **decimal** + " in binary is "

While **decimal** > 0

Compute int **remainder** = **decimal** % 2

Compute int **decimal** = **decimal** / 2

Compute int array **binary** += **remainder**

For (int **i** = **binary.length**) *as long as* (**i** >= 0) *on each iteration* (**i--**)

Print **binary[i]**

If **binary[i]** == 1

Compute int **counter**++

Print "And there are " + **counter** + ", 1s in the binary representation."

End

Trace Tables:

Step #	Input	decimal	remainder	binary	counter	i	Output
0	-5						
1		-5					

Step #	Input	decimal	remainder	binary	counter	i	Output
0	14						
1		14					
2							"14 in binary is "
3		7	0	[0]			
4		3	1	[0],[1]			
5		1	1	[0],[1],[1]			
6		0	1	[0],[1],[1],[1]			
7					1	4	"1"
8					2	3	"1"
9					3	2	"1"
10					4	1	"0"
11							"And there are 4 1s in that binary representation"

Step #	Input	Decimal	remainder	binary	counter	i	Ouput
1	100						
2		100					

Step #	Input	Decimal	remainder	binary	counter	i	Ouput
1	0						
2		0					

Trace Table Notes:

- The algorithm loops if an invalid integer is entered

binary – This is an array of integers, that is used to hold the binary representation of decimal. However since this algorithm prints it backwards, the values in binary will also be stored backwards.

