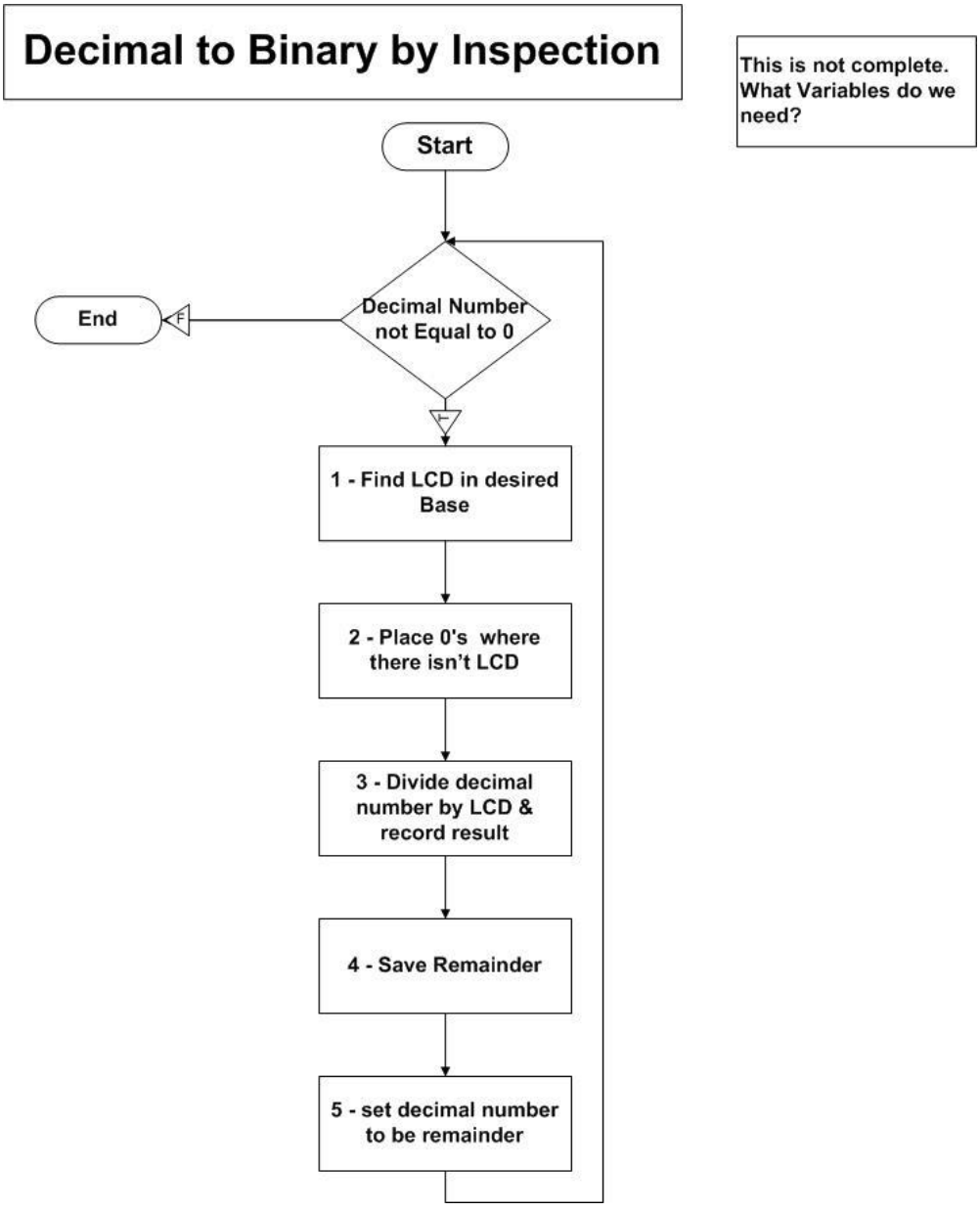
CISS 100 Decimal to Binary Computer Algorithm

Introduction:

Recall that we analyze the inputs, processing and outputs to develop our algorithm. Quite often we will work through the problem and take notes to analyze our thought process. We know how to convert a decimal number to binary by inspection and this suits us for small numbers.

Ex: 19 has 1 - 16, 0 - 8's, 0 – 4's, 1 – 2 and 1 -1.
We did this by successively finding the largest common denominator in binary



What if we want to convert a larger number or automate the process? By inspection how would you convert decimal 1,100,027 to binary? Are you willing to find the largest common divisor in binary?

Is there a better method to encode as an algorithm?

Let's look at an example in decimal:

decimal 957 ➔ decimal

Using our previous inspection method and finding the largest common denominator we see:

9 – 100's, 5 – 10's and 7 – 1's

But we could also successively divide by the base to strip off individual numbers:

957 / 10 ➔ 95 with a remainder of 7
95 /10 ➔ 9 with a remainder of 5
9 / 10 ➔ 0 with a remainder of 9

Decimal to Binary Computer Algorithm

If we analyze the process we recently developed above we see that we are doing the same thing over and over ➔ iteration or a loop. This method is easy to encode in any computer algorithm and can be used to convert from decimal to any base (e.g. binary) using the proper divisor.

Terminology – Let's use the correct mathematical terminology for variable names

Quotient Remainder
Divisor /Dividend

Algorithm:

1. Divide Decimal Number (Dividend) by Divisor (desired base – 2).
2. Save/Record the remainder.moving right to left (In Pseudocode you may just write store remainder)
    a. Remainder obtained using Modulus operator %
    b. Quotient obtained using Integer division operator /
3. Quotient becomes the new Dividend
4. Goto 1.

Work through example (Desk Check) to determine processing for 19. Please create memory spaces for all variables. The computations are below:

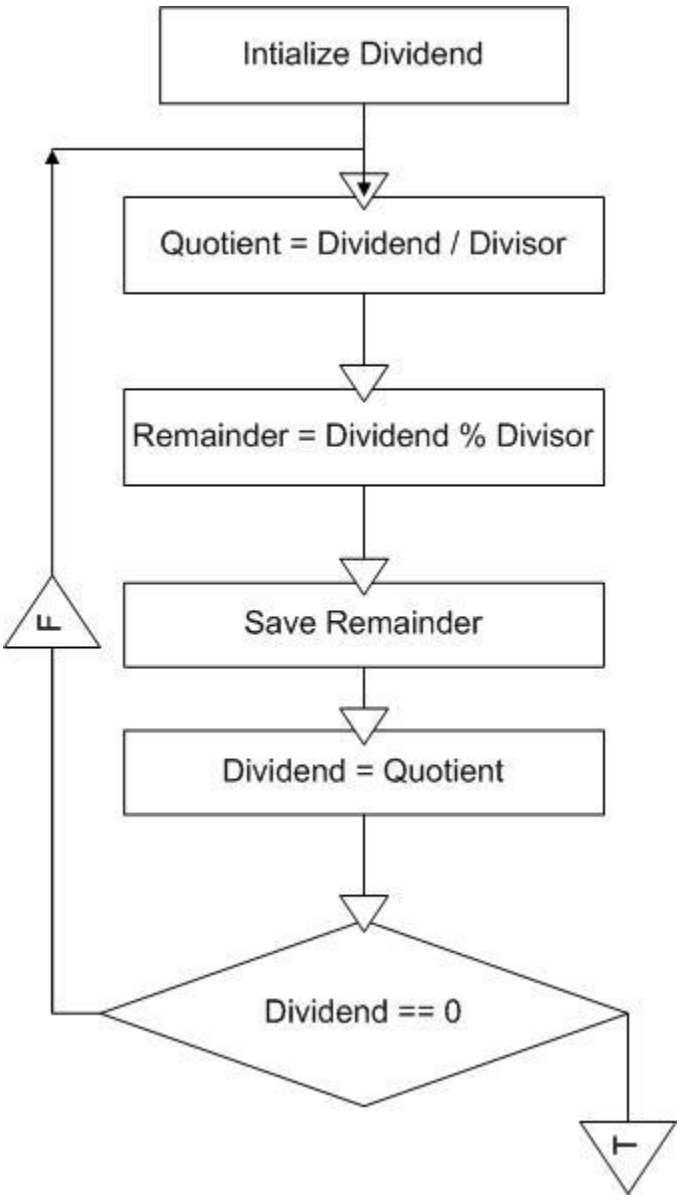
19/2 => 9 r 1
9/2 => 4 r 1
4/2 => 2 r 0
2/2 => 1 r 0
1/2 => 0 r 1 (quotient == 0 – halt/stopping case)

Ans: 10011

Question: What is the stopping case? (Use this stopping case to terminate your loop.)

Question: Is this a counting or Logical loop (determinant or indeterminate)?

Question: Are we testing at the beginning or end of the loop?


Intialize Dividend
Quotient = Dividend / Divisor
Remainder = Dividend % Divisor
Save Remainder
Dividend = Quotient
Dividend == 0
F
T