# Converting Binary to Denary (8bit)

Input Bin  
Loop n from 0 to 7  
 if bin(n) = 1 then  
 number = number + 2^n  
 End if  
end loop  
Output number

# Converting Denary to binary (8bit)

### Subtraction Method

Input Number (between 0 and 255)

Loop n from 7 to 0 stepping -1

BitVal = 2^n

If Number >= BitVal then  
 bin = str(bin + “1”)

Number = number – BitVal

Else

Bin = str(bin + “0”)

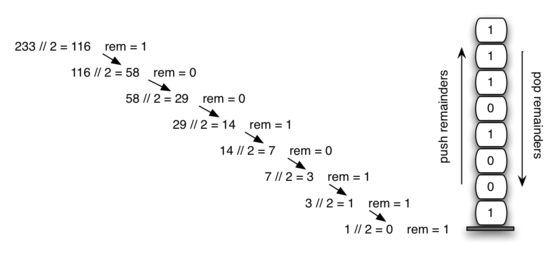
Endif

End Loop

Output Bin

### The Divide by 2 algorithm

The Divide by 2 algorithm assumes that we start with an integer greater than 0. A simple iteration then continually divides the decimal number by 2 and keeps track of the remainder. The first division by 2 gives information as to whether the value is even or odd. An even value will have a remainder of 0. It will have the digit 0 in the ones place. An odd value will have a remainder of 1 and will have the digit 1 in the ones place. We think about building our binary number as a sequence of digits; the first remainder we compute will actually be the last digit in the sequence.



Input Number

do

bin= Str(number Mod 2) + bin

Number = Number / 2)

Loop Until Number < 1

Output bin