

4COM2003.asm-T03L1e02

Given two 8-bit unsigned numbers, X and Y , both greater than 0 and placed at memory locations x and y , use repeated subtraction to divide X by Y , giving two results, Q (the quotient) and R (the remainder), stored at locations $quot$ and $remain$ (where $remain = quot + 1$).

So the relationship $X == (Q * Y) + R$ holds true.

For example, dividing 22 by 6 gives a quotient of 3 and a remainder of 4 ($22 = 6 * 3 + 4$), dividing 10 by 5 gives a quotient of 2 and a remainder of 0 ($10 = 2 * 5 + 0$), and dividing 7 by 12 gives a quotient of 0 and a remainder of 7 ($7 = 0 * 12 + 7$).

The division by repeated subtraction algorithm looks something like this:

```
Q = 0
R = X
while (R >= Y)
{
    R = R - Y
    Q = Q + 1
}
```

Remember: the numbers X and Y are *unsigned* so you will need to use a condition that works with unsigned values. Use the Platforms textbook or the *Conditions and Flags Cheat Sheet* published on the Platforms module site to determine which condition code you should use.

Make sure you test your program with several different pairs of unsigned numbers at locations x and y . (NOTE: The divisor (placed at location y) must never be zero!!)

Fill in the solution template provided in `4COM2003.asm-T03L1e02.asm`, in which we have declared the labels x , y , $quot$ and $remain$ for you.

Calling convention

Leave the address of the result in **r0**. (This has already been coded for you)