102-A04 Computing the Greater Common Divisor (GCD)

Work to do

- You will add the following prototype to **tools.h**
  int gcd (int a , int b);
- You will implement the following function in **tools.c**
- You will invoke this function when menu option 2 is chosen from the **main.c** *main* function.
- You will invoke this function from **tests.c** several times with different parameters and test the return value to make sure it's working completely. Consider this as an implementation of a test-harness that will run automatically.

You need to write a function named *gcd* which computes and returns the greater common divisor of two positive integers *a* and *b* (parameters). The GCD of two positive integers *a* and *b* is computed by assigning a variable *c* to *a* modulo *b*. If *c* is null, then the GCD is *b*, else *a* is assigned the value of *b* and *b* is assigned the value of *c*. The process is repeated until we exit the loop.

If one or both of the parameters are non valid (e.g. negative), your function will simply return -1. The code in the *main* functions in **main.c** and **tests.c** will always check if the call went ok by comparing the return value to -1, it it's equal, the parameters were invalid otherwise the function did its job. Make sure you include such scenario in your test harness.

Example(s)

Here are some examples of return values when calling gcd;

gcd (20 , 10)	→	10
gcd (9 , 3)	→	3
gcd (3 , 9)	→	3
gcd (6 , 9)	→	3
gcd (5 , 9)	→	1
gcd (-1 , 3)	→	-1

Hints

- n/a

Testing

Input		Output	
a	b	Expected	Observed