Computer Science 331

Solutions to Selected Tutorial #2 Questions

Question 3

Requirements specification for `gcd` method

`gcd(int a, int b)`

- pre-condition: a and b are of type `int`
- responsibilities: compute the greatest common divisor of integers a and b using the Euclidean algorithm, where we define

$$\mathrm{gcd}(a,b) = \begin{cases} a & \text{if } b = 0 \\ b & \text{if } a = 0 \\ \mathrm{gcd}(|a|,|b|) & \text{if } a < 0 \text{ or } b < 0 \\ \mathrm{gcd}(a,b) & \text{otherwise} \end{cases}$$

- post-condition: a and b are unmodified
- returns: $\mathrm{gcd}(a,b)$
- exceptions: none (The calling program may verify that the values passed to this function are of type `int`, but the gcd function itself does not need to check this. The type-check that a and b are of type `int` is handled by the compiler, so it is not possible for a and b to be of any other type inside this function.)

Question 4

Develop Black Box Tests

Given just the specifications, it is possible to develop many good boundary and typical test cases, some of which are listed here:

1. Verify that the algorithm works when $a = 0$ (boundary condition)
   - Input: $a = 0$, $b = 31$
   - Expected Output: 31.
2. Verify that the algorithm works when $b = 0$ (boundary condition)
   - Input: $a = 43$, $b = 0$
   - Expected Output: 43.
3. Verify that the algorithm works when $a = 1$ (boundary condition)
   - Input: $a = 1$, $b = 31$
   - Expected Output: 1.
4. Verify that the algorithm works when $b = 1$ (boundary condition)
   - Input: $a = 43$, $b = 1$
   - Expected Output: 1.
5. Verify that the algorithm works when $a = b$ (boundary condition)
   - Input: $a = 31$, $b = 31$
   - Expected Output: 31.
6. Verify that the algorithm works with typical positive inputs with $a > b$ and $\gcd(a, b) = 1$ (typical case)
   - Input: $a = 43$, $b = 31$
   - Expected Output: 1.
7. Verify that the algorithm works with typical positive inputs with $b > a$ and $\gcd(a, b) = 1$ (typical case)
   - Input: $a = 31$, $b = 43$
   - Expected Output: 1.
8. Verify that the algorithm works with typical inputs with $a < 0$, $|a| > b$ and $\gcd(a, b) = 1$ (typical case)
   - Input: $a = -103$, $b = 101$
   - Expected Output: 1.
9. Verify that the algorithm works with typical inputs with $a < 0$, $b > |a|$ and $\gcd(a, b) = 1$ (typical case)

- Input: $a = -101$, $b = 103$
- Expected Output: 1.

10. Verify that the algorithm works with typical inputs with $b < 0$, $a > |b|$ and $\gcd(a, b) = 1$ (typical case)
    - Input: $a = 355$, $b = -134$
    - Expected Output: 1.
11. Verify that the algorithm works with typical inputs with $b < 0$, $|b| > a$ and $\gcd(a, b) = 1$ (typical case)
    - Input: $a = 134$, $b = -355$
    - Expected Output: 1.
12. Verify that the algorithm works with typical negative inputs with $|a| > |b|$ and $\gcd(a, b) = 1$ (typical case)
    - Input: $a = -122$, $b = -99$
    - Expected Output: 1.
13. Verify that the algorithm works with typical negative inputs with $|b| > |a|$ and $\gcd(a, b) = 1$ (typical case)
    - Input: $a = -99$, $b = -122$
    - Expected Output: 1.
14. Verify that the algorithm works with typical positive inputs with $a > b$ and $\gcd(a, b) \neq 1$ (typical case)
    - Input: $a = 781$, $b = 737$
    - Expected Output: 11.
15. Verify that the algorithm works with typical positive inputs with $b > a$ and $\gcd(a, b) \neq 1$ (typical case)
    - Input: $a = 2067$, $b = 23829$
    - Expected Output: 39.
16. Verify that the algorithm works with typical large positive inputs with $b > a$ and $\gcd(a, b) = 1$ (typical case)
    - Input: $a = 1073741827$, $b = 1610612741$
    - Expected Output: 1.

17. Verify that the algorithm works with typical large positive inputs with $a > b$ and $\gcd(a, b) \neq 1$ (typical case)
    - Input: $a = 1342177295$, $b = 1006633185$
    - Expected Output: 5.
18. Verify that the algorithm works with the largest possible positive inputs with $a > b$ (boundary case)
    - Input: $a = 2147483647$, $b = 2147483646$
    - Expected Output: 1.
19. Verify that the algorithm works with the largest possible negative inputs with $|b| > |a|$ (boundary case)
    - Input: $a = -2147483646$, $b = -2147483647$
    - Expected Output: 1.

Question 5

Write Some Code

```
/**
     * Calculates the greatest common divisor of a and b
     *   using the Euclidean algorithm.
     * We define
     *   GCD (0, a) or GCD (a, 0) = a, and
     *   GCD (a, b) = GCD (|a|, |b|) if a < 0 or b < 0.
     *
     * @param a An integer.
     * @param b Another integer.
     * @return GCD(a,b)
     */
public static int gcd (int a, int b)
{
    /* Test if either argument is 0 */
    if (a == 0)
        return (b);
    else if (b == 0)
        return (a);
    else
    {
        /* Take absolute values of a and b */
        int p = Math.abs (a);
        int q = Math.abs (b);

        /* Euclidean algorithm */
        while (q != 0)
        {
            int r = p \% q;
            p = q;
            q = r;
        }

        /* Return GCD */
        return (p);
    }
}
```

Question 6

Develop White Box Tests

Here, we need to ensure that we have test cases for which every line of code will be executed, every execution path will be taken, and multiple possibilities for the number of iterations of the loop will occur. The following test cases, developed by looking at the code listing from the previous question, may be used.

1. The first return statement is executed.
   - Test Case 1 from the black box cases covers this (any test with $a = 0$ will work).
2. The second return statement is executed.
   - Test Case 2 from the black box cases covers this (any test with $b = 0$ will work).
3. The while-loop body executes 0 times.
   - Not possible, because the function would take the second return statement if $q = |b|$ is zero.
4. The while-loop body executes 1 time. (Any inputs for which b divides a will work here.)
   - Input: $a = 35$, $b = 5$
   - Expected Output: 5

5. Cases for which the while-loop body executes multiply (and many) times are covered by the black-box tests.