

Tracing a nested loop

With nested loops, it can at times be difficult to understand what the code is doing. One of the ways to overcome this difficulty is to trace the loop, that is to keep track of how the variables in the conditional statement change with each iteration of the inner and outer loops. Determining how many times each loop executes makes it much easier to understand what the repeated execution of the body of the loop accomplishes. Since the statements in the body of the loop often depend on the loop variables, it would be useful to write down the variables' values for each loop iteration. Here's an example of tracing the values of two variables x and y corresponding to the code segment:

```
int y = 0;
while (y < 10)
{
    int x = 0;
    while (x < 10)
    {
        // some statements
        x++;
    }
    y++;
}
```

Outer loop: the value of y (the current row) is incremented by 1 after all the columns of that row have been processed.

```
y: 0 // y is 0
  x: 0 → 1 → 2 ... → 9 // inner loop iterates 10
                        // times
y: 0 → 1 // y becomes 1
  x: 0 → 1 → 2 ... → 9 // inner loop executes
y: 1 → 2
.
.
y: 8 → 9
  x: 0 → 1 → 2 ... → 9
y: 9 → 10 // loop condition is false
           // loop terminates
```

Inner loop: the value of the x (the current column) goes from 0 to 9 to process all columns in a row. Each execution of the loop starts with x being 0.

As we can see, the inner loop executes 10 times, once for each value of y. Since each inner loop execution involves 10 iterations, the total number of times that the body of the inner loop executes is 10x10 or 100 times.