1. The body of `for (i = a; i <= b; i++)` is iterated $((b+1)-a)/1$ times, or 0 times if $a > b$.

The body of `for (i = a; i <= b; i--)` is iterated forever if $a \leq b$, or 0 times if $a > b$.

The body of `for (i = a; i <= b; i = i+c)` is iterated $((b+1)-a)/c$ times, or 0 times if $a > b$.

2. The condition takes $O(1)$ time and the if-part is empty. Thus the condition has $O(1)$ running time.

3. The condition takes $O(1)$ time and the body is empty. If the condition is false, then we already know it is $O(1)$. If the condition is true then it will repeat forever.

4. The running time of a switch-statement is proportional to the constant number of cases that exist. Each case takes $O(1)$ time, so the sum of all the $O(1)$ terms is $O(1)$ by the summation rule. Each consequent expression takes $O(f_i(n))$ time, so together they take $O(\max(f_1(n), f_2(n), \ldots, f_m(n)))$ time where there are m consequent expressions. We conclude that this big-oh relationship is the running time of a C switch-statement, since we can neglect the $O(1)$ term.

5. The time to evaluate the condition is $O(1)$ and we know it is false. Thus we do not consider the if-part but do consider the else-part. Together the running time of the selection statement is $O(1) + O(g(n))$ which is $O(g(n))$.

6. It takes $O(1)$ time to evaluate the condition, which is false. Hence the body does not execute so we ignore the running time of the body. In total the running time is $O(1)$.