42. Describe the order of growth of each of the following functions using O notation.

a. $N^2 + 3N$

b. $3N^2 + N$

c. $N^5 + 100N^3 + 245$

d. $3N\log_2 N + N^2$

e. $1 + N + N^2 + N^3 + N^4$

f. $(N * (N - 1)) / 2$

a.) $N^2 + 3N$

$N^2 + 3N = O(N^2)$

N^2 is the highest order term.

$3N$ is the lower order term that is ignored.

The constant factor 3 is ignored.

b.) $3N^2 + N$

$= O(3N^2)$

$= O(N^2)$

The highest order term is $3N^2$.

The lower order term N is ignored.

The constant factor 3 is ignored.

c.) $N^5 + 100N^3 + 245$

$= O(N^5 + 100N^3)$

$= O(N^5 + N^3)$

$= O(N^5)$

The highest order term is N^5.

The lower order term $100N^3$ and 245 are ignored.

The constant factor 100 is ignored.

d.) $3N\log_2 N + N^2$

$= O(N\log_2 N + N^2)$

$= O(N^2)$

N^2 is the highest order term.

The constant factor 3 is ignored.

$N\log_2$ is the lower order term to be ignored.

e) $1 + N + N^2 + N^3 + N^4$

$= O(N + N^2 + N^3 + N^4)$

$= O(N^2 + N^3 + N^4)$

$= O(N^3 + N^4)$

$= O(N^4)$

The highest order term is N^4.

The lower order terms to ignore are $1, N, N^2$, and N^3.

The constant factor 1 is ignored.

f.) $(N \cdot (N-1)) / 2$

$= O\left(\frac{N^2 - N}{2}\right)$

$= O\left(\frac{N^2}{2}\right)$

$= O(N^2)$

The highest order term is N^2.

1/2 is the constant factor to ignore as well as N which is the lower order term.

43. Describe the order of growth of each of the following code sections, using O notation:

a.
```
count = 0;
for (i = 1; i <= N; i++)
  count++;
```

b.
```
count = 0;
for (i = 1; i <= N; i++)
  for (j = 1; j <= N; j++)
    count++;
```

c.
```
value = N;
count = 0;
while (value > 1)
{
  value = value / 2;
  count++;
}
```

d.
```
count = 0;
value = N;
value = N * (N - 1);
count = count + value;
```

e.
```
count = 0;
for (i = 1; i <= N; i++)
  count++;
for (i = N; i >= 0; i--)
  count++;
```

f.
```
count = 0;
for (i = 1; i <=N; i++)
  for (j = 1; j <= 5; j++)
    count++;
```

a.) The complexity is O(N). The code goes from 0 to count N. The i value goes from 1 to N incrementing count by 1.

b.) This is a nested loop running from 1 to N. For both the outer and inner loop the is O(N).

c.) This is a time complexity of O(logN). With the value going by 'value/2' the loop runs from value = N.

d.) Since there is no loop, the single statement complexity is O(1).

e.) The first two loops are O(N). The complexity will be added since the 2 loops aren't nested.

'O(N+N)=O(2N)' is the code complexity, while ignoring the constant, 2. The statement is O(N).

f.) There are two nested loops, the outer loop is O(N) while the inner loop is 5(N) since it runs 5 times. Ignore the constant 5. The complexity is O(5·N)=O(N).