This problem set has 12 questions, for a total of 110 points. Answer the questions below and mark your answers in the spaces provided. If the question asks for showing your work, you must provide details on how your answer was calculated.

Your Name: ______________________________

1. For each of the following, give an exact formula $T(n)$ for the number of times the line `// op` is run. Show your work and justify your answer. Assume i increments by 1 at each iteration unless otherwise specified.

   (a) [5 points]

```
for (int i = 0 ; i < 8*n ; i++) {
    // op
}
```

(a) ____________

   (b) [5 points]

```
for (int i = 1 ; i <= n*n*n*n ; i++) {
    // op
}
```

(b) ____________

(c) [5 points]

```
for (int i = 0 ; i < 6*n ; i++) {
    for (int j = 0 ; j < i ; j++) {
        // op
    }
}
```

(c) ______________

(d) [5 points]

```
for (int i = 0 ; i < n*n*n ; i++) {
    for (int j = 0 ; j < i ; j++) {
        // op
    }
}
```

(d) ______________

(e) [5 points]

```
for (int i = 0 ; i < n ; i++) {
    for (int j = 0 ; j < n ; j++) {
        for (int k = 0 ; k < n ; k++) {
            for(int l = 0; l < n; l++) {
                    // op
            }
        }
    }
}
```

(e) ____________

(f) [5 points] Hint: the formula should work with even and odd values of n.

```
for (int i = 0 ; i < n ; i += 2) {   // op   }
```

(f) ____________

(g) [5 points]

```
for (int i = 0 ; i < n ; i += 4) {
    // op
}
```

(g) ______________

(h) [5 points]

```
int m = std::pow(2, n);
for (int i = 1 ; i <= m ; i *= 2) {
    // op
}
```

(h) ______________

(i) [5 points] Hint: Assume n is a power of 2

```
for (int i = n ; i > 1 ; i /= 2) {
    // op
}
```

(i) ______________

2. [5 points] Rewrite the following expression into its closed form (i.e. without the sigma): $\sum_{i=1}^{n}(3+i)$. Show your work.

A. $3+\frac{n*(n+1)}{2}$ B. $3-\frac{n*(n-1)}{2}$ C. $3n-\frac{n*(n+1)}{2}$ D. $3n+\frac{n*(n-1)}{2}$ E. $3n+\frac{n*(n+1)}{2}$

2. ______________

3. [5 points] Rank the following functions by their asymptotic growth rate in **ascending** order.

$\log\log n$ $2^{\log_2 n}$ 2^{300} 4^n $n^2\log n$ $4^{\log_2 n}$

______________ ______________ ______________ ______________ ______________ ______________

4. [10 points] Mark each of the following as true or false.

T(n)	Big O	T/F	Big Omega	T/F	Big Theta	T/F
$\frac{n^3}{10} + 100n \log n$	$O(n \log n)$		$\Omega(n \log n)$		$\Theta(n \log n)$	
$2n^2 + n \log n$	$O(n^2)$		$\Omega(\log n)$		$\Theta(n)$	
$\frac{n}{2} \log n + 4n$	$O(2^n)$		$\Omega(n \log n)$		$\Theta(n \log n)$	
$10\sqrt{n} + 2 \log n$	$O(\log n)$		$\Omega(n)$		$\Theta(\log n)$	
$3\sqrt{n} + 10 \log n$	$O(\sqrt{n})$		$\Omega(1)$		$\Theta(\sqrt{n})$	

5. [10 points] Complete the following table using Big Θ notation with respect to the number of comparisons.

Algorithm	Best Case	Average Case	Worst Case
Selection Sort			
Insertion Sort			
Maximum of an Unsorted Array			
Median of a Sorted Array			
Mode of a Sorted Array			

6. [5 points] Consider the implementation of *insertion-sort* below.

```
void insertion_sort(int *A, int n) {
    for (int i = 0 ; i < n ; i++) {
        print(A, n);
        for (int j = i ; j > 0 ; j --) {
            if (A[j] < A[j-1]) {
                swap(A, j, j-1);
            } else {
                break;
            }
        }
    }
}
```

Given the array A with elements $[19, 32, 64, 18, 5]$ and assuming that print sends the current values of A to the standard ouput. Show what is printed at every iteration of the outer loop.

i = 0					
i = 1					
i = 2					
i = 3					
i = 4					

7. [5 points] Consider the implementation of *selection-sort* below.

```
void selection_sort(int *A, int n) {
    int min_idx;
    for (int i = 0 ; i < n ; i ++) {
        print(A, n);
        min_idx = i;
        for (int j = i+1 ; j < n ; j ++) {
            if (A[j] < A[min_idx]) {
                min_idx = j;
            }
        }
        swap(A, i, min_idx);
    }
}
```

Given the array `A` with elements $[19, 32, 64, 18, 5]$ and assuming that `print` sends the current values of `A` to the standard ouput. Show what is printed at every iteration of the outer loop.

`i = 0`					
`i = 1`					
`i = 2`					
`i = 3`					
`i = 4`					

8. An inversion is any pair of two elements that are out of order. How many inversions are present in each of the following arrays?

   (a) [1 point] $[1, 5, 4, 3, 3, 2]$

   (a) ____________

   (b) [1 point] $[5, 4, 3, 2, 1]$

   (b) ____________

   (c) [1 point] $[1, 2, 3, 4, 5]$

   (c) ____________

   (d) [1 point] $[5, 1, 4, 2, 4]$

   (d) ____________

   (e) [1 point] $[6, 9, 1, 4, 10]$

   (e) ____________

9. Consider the following segments of code:

   (a) [5 points] Give the exact number of multiplications $T(n)$

```
int s = 0;
for (int i = 0; i < 5*n; i++) {
        for(int j = 0; j < i; j++) {
                s = s * i;
        }
}
```

   (a) ____________

   (b) [5 points] Give the exact number of multiplications $T(n)$

```
int s = 0;
for (int i = 0; i < 5*n; i++) {
        for(int j = 0; j < i; j++) {
                s = s * i;
        }
        s = s * 2;
}
```

   (b) ____________

10. Consider the following functions.

```
int foo(int x, int *y) {
    x = x + 20;
    *y = x * 3;
    return x;
}

int *bar(int x) {
    int y = 100 + x;
    return &y;
}
```

For each of the questions below, what are the values of `x` and `y` after running the provided line of code in the form **x,y**. If you think the code may trigger an error at any point indicate the reason. **Do not use a computer for solving this question.**

(a) [2 points] `int x = 3, y = 4; x = foo(x, &y);`

(a) ______________

(b) [2 points] `int x = 20, y = 30; x = foo(x, &y);`

(b) ______________

(c) [2 points] `int x = 0, y = 0; x = foo(x, &y);`

(c) ______________

(d) [2 points] `int x = 1, y = 3; int *z = bar(y); x = *z;`

(d) ______________

(e) [2 points] `int x = 1, y = 0; int *z = bar(y); x = *z;`

(e) ______________

Optional Questions

The following questions are **optional**, and will not be graded. They are simple problems that should serve as decent practice for solving problems on paper.

11. Write an algorithm in $\Theta(n)$ time that calculates the missing number in an array A of integers. The length of the array is $n-1$ and every element $A[i]$ in the array is such that $1 \leq A[i] \leq n$. For example, given $A = [3, 2, 1, 5]$ the output sould be 4.

12. Write an algorithm that removes all duplicate integers from an input array A. The length of A is n and every element $A[i]$ is such that $1 \leq A[i] \leq 100$. For example, given $A = [12, 2, 2, 3, 4, 2, 5]$, the algorithm should return $[12, 2, 3, 4, 5]$. Can you make your algorithm run in $\Theta(n)$ time?