Instructions: This homework assignment covers some of the later course topics: Recursion, Stacks, Queues and Algorithm Analysis. The answers may be determined by reading the CS 1704 course notes and experimenting with code.

The quizzes will be submitted on-line through the Computer Science Curator server. No late submissions will be accepted.

For the next four questions, apply the rules given in the course notes and in class to determine the most appropriate (tightest) big-O classification of the given function. All logarithms are base 2.

1. \( f(n) = 43n \log n + 3n^2 \) is \( O(\underline{\text{______}}) \)
   
   1) \( O(1) \)  
   2) \( O(\log n) \)  
   3) \( O(n) \)  
   4) \( O(n \log n) \)  
   5) \( O(n^2) \)  
   6) \( O(n^2 \log n) \)  
   7) \( O(2^n) \)  
   8) None of these

2. \( f(n) = 9n + 23 \log n \) is \( O(\underline{\text{______}}) \)
   
   1) \( O(1) \)  
   2) \( O(\log n) \)  
   3) \( O(n) \)  
   4) \( O(n \log n) \)  
   5) \( O(n^2) \)  
   6) \( O(n^2 \log n) \)  
   7) \( O(2^n) \)  
   8) None of these

3. \( f(n) = 17 \log n^2 \) is \( O(\underline{\text{______}}) \) // careful here...
   
   1) \( O(1) \)  
   2) \( O(\log n) \)  
   3) \( O(n) \)  
   4) \( O(n \log n) \)  
   5) \( O(n^2) \)  
   6) \( O(n^2 \log n) \)  
   7) \( O(2^n) \)  
   8) None of these

4. \( f(n) = 3n(\log n + 7n) \) is \( O(\underline{\text{______}}) \)
   
   1) \( O(1) \)  
   2) \( O(\log n) \)  
   3) \( O(n) \)  
   4) \( O(n \log n) \)  
   5) \( O(n^2) \)  
   6) \( O(n^2 \log n) \)  
   7) \( O(2^n) \)  
   8) None of these

For questions 5 through 8, assume that \( N \) is an integer variable with a positive value, and any other declarations necessary to make the given code fragment syntactically and logically correct.

5. Execution of the following code fragment is \( O(\underline{\text{__________}}) \).

   ```java
   for (int Idx = 0; Idx < N; Idx++) {
       System.out.println( A[Idx] );
   }
   ```

   1) \( O(1) \)  
   2) \( O(\log N) \)  
   3) \( O(N) \)  
   4) \( O(N \log N) \)  
   5) \( O(N^2) \)  
   6) None of these
6. Execution of the following code fragment is $O(\text{__________}).$

```java
int i = N;
while (i > 0) {
    int Sum = 0;
    for (int j = 0; j < N; j++)
        Sum++;
    System.out.println(Sum);
    i--;
}
```

1) $O(1)$  2) $O(\log N)$  3) $O(N)$  4) $O(N \log N)$  5) $O(N^2)$  6) None of these

7. Execution of the following code fragment is $O(\text{__________}).$

```java
int i = N;
while (i > 0) {
    int Sum = 0;
    for (int j = 0; j < i; j++)
        Sum++;
    System.out.println(Sum);
    i--;
}
```

1) $O(1)$  2) $O(\log N)$  3) $O(N)$  4) $O(N \log N)$  5) $O(N^2)$  6) None of these

8. Execution of the following code fragment is $O(\text{__________}).$ (Be careful, tricky.)

```java
int Max = N;
int Sum = 0;
for (int i = 1; i <= Max; i++) {
    for (int j = 1; j <= Max; j = 2*j)
        Sum++;
}
```

1) $O(1)$  2) $O(\log N)$  3) $O(N)$  4) $O(N \log N)$  5) $O(N^2)$  6) None of these
9. Execution of the following code fragment is $O(\underline{\text{\hspace{2cm}}})$.

```c
int Max = N;
int Sum = 0;
Sum += Max * Max;
```

1) $O(1)$  
2) $O(\log N)$  
3) $O(N)$  
4) $O(N \log N)$  
5) $O(N^2)$  
6) None of these

10. True or False. For medium sized problems of approximately 1000 data elements $N^2$ algorithms can be used in a program.

1) True  
2) False