Assignment #1 –
CS2001-Data Structures
Fall 2024
Deadline: *16th September 2024 (5:30 PM in Class)*

Instructions:

1. **Guidelines Compliance**: *Any violation of the provided guidelines will result in a score of zero for the assignment.*
2. **Handwritten Submission**: *The assignment must be completed in your own handwriting.*
3. **Individual Work**: *This is an individual assignment. Collaboration or copying will not be tolerated.*
4. **No Late Submissions***: **Late submissions will not be accepted under any circumstances.***
5. **Identification Details**: *Ensure that your Roll Number, Section, Name, and Signature are clearly written at the bottom of each page. Failure to do so will result in mark deductions, and no subsequent queries regarding this matter will be entertained.*
6. **Code Complexity Analysis:**
   - *Perform a detailed **complexity analysis** for each provided code segment.*
   - *Include **Time Complexity** for each code.*
   - *Break down the complexity step-by-step, including the analysis of loops, and any nested structures.*
   - *If there are multiple solutions or approaches, discuss the complexities of each and provide a rationale for which approach is more efficient.*
7. **Clarity and Neatness:**
   - *Your handwriting must be legible and neat. Any illegible or messy submissions may result in mark deductions.*
8. **Reference Material:**
   - *While you may consult textbooks or other resources for your understanding, ensure all work is done in your own words.*
9. **Additional Sheets:**
   - *You may attach extra sheets if required to complete your analysis or explanations. Ensure that any additional sheets also contain your Roll Number, Section, Name, and Signature at the bottom.*

Note:
Please adhere strictly to these instructions to ensure your assignment is graded fairly. Good luck and put forth your best effort in understanding and analyzing the complexities of the algorithms.

Roll Number _____ Section_____ Name_____ Signature_____

Find the complexity of the following codes, show all steps.

1.
```
for i = 1 to n:
{
   for j = i to n
   {
      sum += a[i][j]
   }
}
```

2.
```
while (n > 0)
{
   ans += n;
   n /= 2;
}
```

3.
```
for (i = n; i >= 1; i /= 2):
{
   for j = m to i
   {
      ans += (i * j);
   }
}
```

4.
```
cout << "Hello world";
```

5.
```
for i = 1 to n
{
   for j = n to i
   {
      ans += (i * j);
   }
}
```

6.
```
for( i = 0; i < n; i++)
{
   for( j = 0; j < 5; j *= 2)
      statements; // worth O(1)
}
```

7.
```
for( i = n; i >= 0; i--)
{
   for( j = 0; j < n; j++)
      statements; // worth O(1)
}
```

8.
```
// n > 0
k = 0;
while(1)
{
   for(i = 0; i < n; i++)
      continue;

   if(k == !n)
      break;
}
```

9.
```
// n > 0
do
{
   n--;
} while(n < 0);
```

10.
```
int operator ++(int &n)
{
   for(int i = 1; i <= x; i++)
   {
      cout << "*";
   }
   n++;
   return n;
}
int main()
{
   for(int i = 0; i < n; ++i)
   {
      continue;
   }
}
```

11.
```
sum = 0;
for( i = 1; i <= n * n; i++ )
   sum++;
```

12.
```
sum = 0;
for( i = 1; i < n; i++ )
( j = 1; j < i * n; j++ )
if( j % 1 == 0 )
( k = 0; k < j; k++ )
```

13.
```
sum = 0;
for( i = 1; i < n; i++ )
   for( j = 0; j < i * n; j++ )
      ( k = 0; k < j; k++)
         sum++;
```

14.
```
sum = 0;
for( i = 1; i <= n; i++ )
   for( j = 0; j < i; j++ )
      sum++;
```

15.
```
sum = 0;
for( i = 1; i <= n * n; i++ )
   for( j = 0; j < n; j = j + 1 )
      sum++;
```

16.
```
sum = 0;
for( i = 1; i < n; i = i * 2 )
   for( j = 0; j < n; j++ )
      sum++;
```

17.
```
void fun(int n)
{
   int I, j, k, count = 0;
   for(i = n / 2; i <= n; i++)
      for(j = 1; j + n / 2 <= n; j++)
         for(k = 1; k <= n; k = k * 2)
            count++;
}
```

18.
```
Sum()
{
   for i = 1 to n do
      i = i + 1;
   for j = 1 to n do
      for k = n down to 1 do
         sum = sum + 1;
}
```

Roll Number _____ Section_____ Name_____ Signature_____

19.
```
int a = 0;
for (i = 0; i < N; i++) {
   for (j = N; j > i; j--) {
      a = a + i + j;
   }
}
```

20.
```
for(i = n ; i > 0; i--)
{
   for(j = 0; j < n; j++)
   {
      cout << i;
   }
}
```

21.
```
for(i = n ; i > 0; i--)
{
   for(j = 0; j < n; j * 2)
   {
      cout << i;
   }
}
```

22.
```
int total = 0;
for (int i = 1; i <= n * n; i++)
{
   total++;
}
```

23.
```
int m = 0;
for (int i = 1; i <= n; i *= 2)
{
   for (int j = 0; j < n; j++)
   {
      m++;
   }
}
```

24.
```
int a = 0;
for (i = 0; i < n; i++)
{
   for (j = n; j > i; j--)
   {
      a = a + i + j;
   }
}
```

25.
```
while(low <= high)
{
   mid = (low + high) / 2;
   if (target < list[mid])
      high = mid - 1;
   else if (target > list[mid])
      low = mid + 1;
   else break;
}
```

26.
```
int key, j;
for(int i = 1; i < size; i++)
{
   key = array[i];
   j = i;
 while(j > 0 && array[j-1] > key)
   {
      array[j] = array[j-1];
      j--;
   }
   array[j] = key;
}
```

27.
```
int sum = 0;
for (int i = 1; i <= n * n; ++i)
{
   for (int j = 0; j < n; ++j)
   {
      ++sum;
   }
}
```

28.
```
int i, j, imin;
for(i = 0; i < size-1; i++)
{
   imin = i;
   for(j = i + 1; j < size; j++)
   {
      if(array[j] < array[imin])
         imin = j;
   }
   swap(array[i], array[imin]);
}
```

29.
```
for(i = 0; i < n; i++)
{
   i *= 3;
}
```

30.
```
long double sum = 0;
for (int i = 1; i <= n; ++i)
{
   for (int j = 1; j < i * n; ++j)
   {
      if (j % 1 == 0)
      {
         for (int k = 0; k < j; ++k)
         {
            ++sum;
         }
      }
   }
}
```

31.
```
int sum = 0;
for (int i = 0; i <= n; ++i)
{
   for (int j = 0; j < m; ++j)
   {
      ++sum;
   }
}
```

Roll Number _____ Section_____ Name_____ Signature_____

32.

```
for ( i = 0 ; i < n ; i++ )
{
    m += j;
    m += j;
    m += j;
    …
    m += j;   // 31 times
}
```

33.

```
for ( i = 0 ; i < n ; i++ ) {
   subtotal = 0;
   for( j = 0 ; j < i; j++)
      subtotal += j;
   tot += subtotal;
}
```

34.

```
for ( i = 0 ; i < n ; i++ )
   for( j = 0 ; j < sqrt(995) ; j++ )
      m += j;
```

35.

```
for ( i = 0 ; i < n ; i++ )
   for( j = 0 ; j < sqrt(n) ; j++ )
      m += j;
```

Roll Number _____ Section_____ Name_____ Signature_____