DSA-Lab 4-Time Complexity

Total Grade Points - 100

1. Arithmetic Sequence Size

(There are many terms in this sequence?)

1.1. $\Omega(N) = N/2 \le \|\{1, 2, 3, 4, 5, 6, \ldots, N\}\| \le N = O(N)$ # of terms are $\Theta(N)$	for(int i=1; i<=N; i++);
1.2. $\Omega(N) = N/4 <= \|\{0, 2, 4, 6, 8, 10, \ldots, N\}\| \le N/2 = O(N)$ # of terms are $\Theta(N)$	for(int i=0; i<=N; i+=2);
1.3. $\Omega(N) = N/4 <= \|\{1, 3, 5, 7, \ldots, N\}\| \le N/2$ i.e $O(N)$ many terms # of terms are $\Theta(N)$	for(int i=1; i<=N; i+=2);
1.4. $\Omega(N) = N/6 <= \|\{1, 4, 7, 10, \ldots, N\}\| \le N/3$ i.e. $O(N)$ # of terms are $\Theta(N)$	for(int i=1; i<=N; i+=3);
1.5. $\Omega(N) = \|\{1, 1+k, 1+2k, 1+3k, 1+4k, 1+5k, \ldots, N\}\| \le N/k$ i.e. $O(N)$ if k is a constant	for(int i=1; i<=N; i+=k);
1.6. $\Omega(N/\log N) = \|\{1, 1+\log N, 1+2\log N, 1+3\log N, 1+4\log N, 1+5\log N, \ldots, N\}\| \le N/\log N$ i.e. $O(N/\log N)$	K = log N; for(int i=1; i<=N; i+=k);
1.7. $\Omega(\sqrt{N}) = \|\{1, 1+\sqrt{N}, 1+2\sqrt{N}, 1+3\sqrt{N}, 1+4\sqrt{N}, 1+5\sqrt{N}, \ldots, N\}\| \le N/\sqrt{N} = O(\sqrt{N})$ i.e. $\Theta(\sqrt{N})$	K = $\sqrt{N}$; for(int i=1; i<=N; i+=k);
(int i=1; i<=N; i+=10); N/10 times Similarly for(int i=1; i<=N; i+=20); N/20 times for(int i=1; i<=N; i+= $\sqrt{N}$); $N/\sqrt{N} = \sqrt{N}$ ====> $N = \sqrt{N} . \sqrt{N}$	

---

2. Arithmetic Series and relatives

Applications of $1+2+3+4+\ldots+N = \frac{N(N+1)}{2}$ If you don't remember this formula.

Proof

Upper Bound	1+2+3+4+5+6+ ....(N-3)+ (N-2)+ (N-1)+ N/2 <= N + N + N + ... + N <= N x N = N^2 = O(N^2)
Lower Bound	~~1+2+3+4+5+6+...+ N/2~~+ (N/2+1)+(N/2+2) ....(N-3)+ (N-2)+ (N-1)+ N >= (N/2+1)+(N/2+2) + ... +(N-3)+ (N-2)+ (N-1)+ N >= N/2 + N/2 + N/2 + N/2 + ... + N/2 + N/2 + N/2 = N/2 x N/2 = ¼ N^2 = $\Omega(N^2)$ f(N) = Θ(N^2)

2.1 $\Omega(T^2) \le 1+2+3+4+5+6+ \ldots T-3+ T-2+ T-1+ T \le O(T^2)$ ====> $\Theta(T^2)$	
2.2 $\Omega(N^2) \le 1+2+3+4+5+6+ + \ldots + N/2+N/2+1+ \ldots +N-3+ N-2+ N-1+ N \le O(N^2)$ ===> $\Theta(N^2)$ $\Theta(N)$ $_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\ldots\ldots+_{(1,2,3,4\ldots,N)}N$ $\Theta(N^2)$	for(int i=1; i<=N; i++) for(int j=1; j<=i; j++)
2.3 $\Omega(N^2) \le 1+2+3+4+5+6+ \ldots(N/2-3)+ (N/2-2)+ (N/2-1)+ N/2 \le O(N^2)$ ====> $\Theta(N^2)$ $\Theta(N)$ $_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\ldots\ldots+_{(1,2,3,4\ldots,N)}N$ $\Theta(N^2)$	for(int i=1; i<=N; i+=2) for(int j=1; j<=i; j++)
2.4 $\Omega(N^2) \le 1+2+3+4+5+6+ \ldots(N/3-3)+ (N/3-2)+ (N/3-1)+ N/3 \le O(N^2)$ ====> $\Theta(N^2)$ $\Theta(N)$ $_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\ldots\ldots+_{(1,2,3,4\ldots,N/3)}N/3$ $\Theta(N^2)$	for(int i=1; i<=N; i+=3) for(int j=1; j<=i; j++)
2.5 $\Omega(N) \le 1+2+3+4+5+6+ \ldots + \sqrt{N}$ <= $O((\sqrt{N})^2)$ $\le O(N)$ ====> $\Theta(N)$ $\Theta(N^{1/2})$ $_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\ldots\ldots+_{(1,2,3,4\ldots,N^{1/2})}N^{1/2}$ $\Theta(N)$	for(int i=1; i<=$N^{1/2}$; i+=1) for(int j=1; j<=i; j++)
2.6 $\Omega((\log N)^2) \le 1+2+3+4+5+6+ \ldots + \log N$ <= $O((\log N)^2) \le \Theta(\log^2 N)$	for(int i=1; i<=N; i*=2) $\Theta(\log N)$

$_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\dots\dots+_{(1,2,4,8\dots,N)}\log N = \Theta(\log^2 N) \Longrightarrow$
$_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\dots\dots+_{(1,2,4,8\dots,N)}\log N$
	for(int j=1; j<=i; j*=2)
	<u>Example 2</u>
	for(int i=1; i<=log N; i++)
	for(j=1; j<=i; j++) ;

2.7 $\Omega(N^4) \leq 1+2+3+4+5+6+ \dots + N^2 \leq O(N^4) \Longrightarrow \Theta(N^4)$ $_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\dots\dots+_{(1,2,3,4\dots,N^2)}N^2 = \Theta(N^4) \Longrightarrow$	for(int i=1; i<=N*N; i=1) $\Theta(N^2)$ for(int j=1; j<=i; j++)
2.8 $\Omega(N^6) \leq 1+2+3+4+5+6+ \dots + N^3 \leq O(N^6)$ $\Theta(N^3)$ $_{(1)}1+_{(1,2)}2+_{(1,2,3)}3+_{(1,2,3,4)}4+\dots\dots+_{(1,2,3,4\dots,N^3)}N^3$ $\Theta(N^6)$	for(int i=1; i<=N*N*N; i=1) for(int j=1; j<=i; j++)
2.9 $\Omega(N^{2k}) \leq 1+2+3+4+5+6+ \dots +N^k <= O(N^k x N^k)$	
2.10 $\Omega(N^3) \leq 1^2+2^2+3^2+4^2+5^2+6^2+ \dots + N^2 \leq O(N^3)$ $_{(1)}1+_{(1,2,3,4)}2+_{(1,2,3,\dots,9)}9+_{(1,2,3,4\dots,16)}16+\dots\dots+_{(1,2,3,4\dots,N^2)}N^2 = \Theta(N^3)$	for(int i=1; i<=N; i=1) $\Theta(N)$ for(int j=1; j<=i*i; j++)
2.11 $\Omega(N^4) \leq 1+2^3+3^3+4^3+5^3+6^3+ \dots + N^3 \leq O(N^4)$ $\Theta(N)$ $_{(1)}1+_{(1,2,3,\dots,8)}8+_{(1,2,3,\dots,27)}27+_{(1,2,3,4\dots,64)}64+\dots\dots+_{(1,2,3,4\dots,N^3)}N^3$ $\Theta(N^4)$	for(int i=1; i<=N; i=1) for(int j=1; j<=i*i*i; j++)
2.12 $\Omega(N^{k+1}) \leq 1^k+2^k+3^k+4^k+5^k+6^k+ \dots +N^k <= O(N^{k+1})$	

3. Some Examples

$$\sqrt{N} * \sqrt{N} = N$$

for(int i=1; i*i<=N; i++) Sum++; $O(\sqrt{N})$ for(int i=1; i*i<=N*N; i++) Sum++; O(N)

for(int i=1; i*i*i<=N*N; i++) Sum++; $O(N^{2/3})$ for(int i=1; i*i*i<=N; i++) Sum++; $O(N^{1/3})$

4. Geometric Sequence Size

4.1. $\lvert\{N, N/2, N/4, N/8, N/2^4, N/2^5, N/2^6, \dots 8, 4, 2, 1\}\rvert = \log_2 N$
for(int i=1; i<=N; i*=2) or for(int i=N; i*i>=1; i/=2)

4.2. $\lvert\{N, N/3, N/9, N/27, N/3^4, N/3^5, N/3^6, \dots, 3^3, 9, 3, 1\}\rvert = \log_3 N$
for(int i=1; i<=N; i*=3) or for(int i=N; i*i>=1; i/=3)

$\boxed{O(\log N)}$

4.3. $\lvert\{N, N/5, N/25, N/125, N/5^4, N/5^5, N/5^6, \dots, 5^3, 5^2, 5, 1\}\rvert = \log_5 N$
for(int i=1; i<=N; i*=5) or or for(int i=N; i*i>=1; i/=5)

4.4. $\lvert\{N, N/k, N/k^2, N/k^3, N/k^4, N/k^5, N/k^6, \dots, k^3, k^2, k, 1\}\rvert = \log_k N$
for(int i=1; i<=N; i*=2) or or for(int i=N; i*i>=1; i/=k)

Some formulas,

1. $\log N^k = k \log N$	2. $\log N^2 = 2 \log N = \log N + \log N$	3. $\log^2 N = \log N \cdot \log N$

5. GEOMETRIC SERIES O(by the largest term)

Any Geometric Series with multiplicand factor of greater than 2(if increasing) or smaller then ½ (for decreasing geometric series) is bounded above and below by the largest term. If it is an increasing Geometric series in that case it will be the last term, if it is a decreasing series it will be the first term. For any constant - ratio(multiplication factor greater than 2 the above inequality is valid).

5.1. $\Omega(N) = N < 1+2+4+8+16+32+\ldots +N/4+N/2+N < 2N = O(N)$ ==> $\Theta(N)$

```
for(int i=1; i<=N; i*=2)
        for(int j=1; j<=i; j++)
```

5.2. $\Omega(N) = N < 1+3+9+3^3+3^4+3^5+\ldots +N/3^2+N/3+N < 2N = O(N)$==> $\Theta(N)$

```
for(int i=1; i<=N; i*=3)
        for(int j=1; j<=i; j++)
```

5.3. $\Omega(N^2) = N^2 < 1+3+9+3^3+3^4+3^5+\ldots +N^2/3^2+N^2/3 + N^2 < 2N^2 = O(N^2)$==>$\Theta(N^2)$

```
for(int i=1; i<=N*N; i*=5)
        for(int j=1; j<=i; j++)
```

5.3. $N^3 < 1+5+5^2+5^3+5^4+5^5+\ldots +N^3/5^2+N^3/5+N^3 < 2N^3 = O(N^3)$

$1+5+5^2+5^3+5^4+5^5+\ldots +N^3/5^2+N^3/5+N^3 < 2N^3$

```
for(int i=1; i<=N*N*N; i*=5)
        for(int j=1; j<=i; j++)
```

5.4. $N^{3/2} < 1+5+5^2+5^3+5^4+5^5+\ldots +N^{3/2}/5^2+N^{3/2}/5+N^{3/2} < 2N^{3/2}$

```
for(int i=1; i<=N^(3/2); i*=2 )
        for(int j=1; j<=i; j++)
```

5.5. $N^{1/2} < 1+5+5^2+5^3+5^4+5^5+\ldots +N^{1/2}/5^2+N^{1/2}/5+N^{1/2} < 2N^{1/2}$

```
for(int i=1; i<=N^(1/2); i*=2 )
        for(int j=1; j<=i; j++)
```

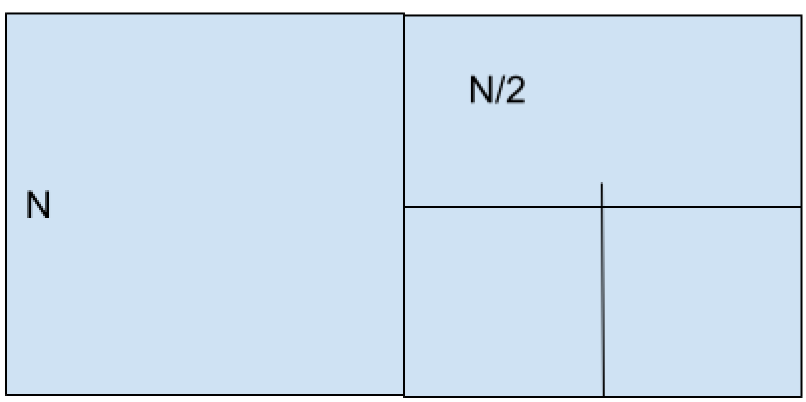
5.6. $N < 1+5+5^2+5^3+5^4+5^5+\ldots +N/5^2+N/5+N < 2N$

$\log_a N = \frac{\log_b N}{\log_b a}$ => $\log_2 N = \log_{10} N / \log_{10} 2$ => $\log_2 N = 1/\log_{10}2 \,.\, \log_{10} N$

Implies => $\log_a N = \log_b N$

Proof of Geometric Series?

$\Omega(N) = N < 1+2+4+8+16+32+\ldots +N/4+N/2+N < 2N = O(N)$ ==> $\Theta(N)$



Question 1

(Estimated Time 20 mins) **(Each proof contains 5 points)**

Prove all the thetas (using halving technique or any way you like) . Both find out its lower(Omega) and upper(Big O) bound by the same function so that you can declare it to be theta of the given function.

1. **$1^2 + 2^2 + 3^2 + 4^2 + \ldots\ldots + N^2$ -- $\Theta(N^3)$**

2. **$1 + 2 + 3 + 4 + \ldots\ldots\ldots + N^2$ -- $\Theta(N^4)$**

3. **$1 + 3 + 5 + 7 + 9 + \ldots\ldots\ldots + (2N + 1)$ -- $\Theta(N^2)$**

4. **$2 + 4 + 6 + 8 + \ldots\ldots\ldots + 2N$ -- $\Theta(N^2)$**

5. **$1 + 2 + 3 + 4 + \ldots\ldots\ldots + (N/2)$ -- $\Theta(N^2)$**

6. **$1+2+4+8+16+ \ldots + N^2$ ---- $\Theta(N^2)$**

Question 2

(Estimated Time 100 mins)

(35*2 = 70 Points)

What is the algorithm's complexity of the following piece of code - Sample Solution is in RED.
 	
int Sum=0; // O(1) Time
for(int i=0; i<N; i++) //(1+1+1+...+1 - - - N Times =O(N)
for(int j=0; j<N; j++) Sum++;
// (1+1+1+...+1) + (1+1 +... +1)+... + (1+1 +... +1) added N times
// N + N +... + N = $O(N^2)$
Overall Complexity : $O(1) + O(N) + O(N^2) + O(N^2) = O(N^2)$
	2) What is the algorithm's complexity of the following piece of code
	
	int Sum=0;
	for(int i=0; i<N; i++)
	Sum++;
	for(int j=0; j<N; j++)
	Sum++;

3)
 	
What is the algorithm's complexity of the following piece of code
 	
int Sum=0;
for(int i=0; i<N; i++)
for(int j=0; j<N; j++)
for(int k=0; k<N; k++)
Sum++;
 	
for(int i=0; i<N; i++)
for(int j=0; j<N; j++)
for(int k=0; k<N; k++)
Sum++;
	4)
	What is the algorithm's complexity of the following piece of code
	
	int Sum=0;
	for(int i=0; i<N; i++)
	Sum++;
	for(int j=0; j<N; j++)
	Sum++;
	for(int k=0; k<N; k++)
	Sum++;
	for(int m=0; m<N; m++)
	Sum++;
	for(int n=0; n<N; n++)
	Sum++;
	for(int p=0; p<N; p++)
	Sum++;

5)
 	
int Sum=0;
for(int i=0; i<N; i++)
for(int j=0; j<i; j++)
for(int k=0; k<j; k++)
Sum++;
	6
	
	int Sum=0;
	for(int i=0; i<N; i+=2)
	for(int j=0; j<i; j+=2)
	for(int k=0; k<j; k+=2)
	Sum++;

7
int Sum=0;
for(int i=1; i<N; i*=2)
for(int j=1; j<N; j*=2)
Sum++;
	8
	int Sum=0;
	for(int i=1; i<N; i*=2)
	Sum++;
	for(int j=1; j<N; j*=2)
	Sum++;

9

```
for(int i=1; i<=N*N; i+=2)
   for(int j=1; j<N*N; j*=2)
        Sum++;
```

10

```
for(int i=1; i<=N*N; i+=2)
  Sum++;

for(int j=1; j<N*N; j*=2)
        Sum++;
```

11

```
for(int i=1; i<=N*N; i*=2)
   for(int j=1; j<N*N; j*=2)
        Sum++;
```

12

```
for(int i=1; i<=N*N; i*=2)
        Sum++;

for(int j=1; j<N*N; j*=2)
        Sum++;
```

13

```
int Sum=0;
for(int i=1; i<=N; i*=2)
 for(int j=1; j<=N; j*=2)
   for(int k=1; k<=N; k*=2)
        Sum++;
```

14

```
int Sum=0;
for(int i=1; i<=N; i*=2)
Sum++;
for(int j=1; j<=N; j*=2)
        Sum++;
for(int k=1; k<=N; k*=2)
        Sum++;
```

15

```
            int sum,i,j;
            sum = 0;
            for (i=1;i<n;i=i*2)
            {
              for (j=0;j<n;++j)
              {
                     sum++;
               }
            }
```

16

<u>BE CAREFUL GEOMETRIC SERIES</u>

```
int sum,i,j;
sum = 0;
for (i=1; i<n; i=i*2)
{
  for (j=0; j <i ; ++j)
  {
         sum++;
   }
}
```

17

<u>BE CAREFUL GEOMETRIC SERIES</u>

```
int sum,i,j;
sum = 0;
for (i=1; i<n; i=i*5)
{
  for (j=0; j<i; j+=2)
  {
         sum++;
   }
}
```

18

```
int sum,i,j;
sum = 0;
for (i=1; i<n;  i=i*4)
{
  for (j=0 ; j<n ; j+=3)
  {
         sum++;
   }
}
```

19 What will be the output (the value of Sum) of the program asymptotically in BIG-O notation, I am not asking here the complexity of loop rather the asymptotic bound on the value of Sum:

```
int Sum = 0;
for(int i=1; i<=n; i+=1)
{
  Sum+=i;
}
cout<<Sum<<endl;
```

20 What will be the output(the value of Sum) of the program asymptotically in BIG-O notation:

```
int Sum = 0;
for(int i=1; i<=n; i*=2)
{
  Sum+=i;
}
cout<<Sum<<endl;
```

21 What is the time complexity of the algorithm:

```
int Sum = 0;
for(int i=1; i<=n; i+=1)
{
  for(int j=1; j<=i; j++)
  {
      Sum++;
  }
}
cout<<Sum<<endl;
```

22 What is the time complexity of the algorithm:

```
int Sum = 0;
for(int i=1; i<n; i*=2)
{
  for(int j=1;  j<=i; j++)
  {
      Sum++;
  }
}
cout<<Sum<<endl;
}
```

40* Complexity of primeNumber function.

```
int sqrt(int N)
{
  int d;
  for(d=0; d*d<=N; d++) {   }
  return d-1;
}
bool primeNumber(int n)
{
        bool isPrime = true;
        int lmt = (sqrt(n));
        for (int d=2; d <=lmt ;++d)
        {
                if (n%d==0)
                        return false;
        }
        return true;
}
```

41* Complexity of primeNumber function.

```
int sqrt(int N)
{
  int d;
  for(d=0; d*d<=N; d++){ }
  return d-1;
}
bool primeNumber(int n)
{
        bool isPrime = true;
        for (int d=2; d <= sqrt(n) ;++d)
        {
                if (n%d==0)
                        return false;
        }
        return true;
}
```

23 What is the time complexity of the algorithm:

```
int f1(int n)
{
        int K=0;
        for(int j=0; j*j<=n*n; j++)  K++;
        return K;
}
int main()
{
        int Sum = 0,  n;
        cin>>n;
        for(int i=1; i<=f1(n); i+=1)
          for(int j=1; j<=i; j++)    Sum++;
        cout<<Sum<<endl;
}
```

24

What is the time complexity of the algorithm:

```
int f1(int n)
{
        int K=0;
        for(int j=1; j*j<=n; j*=2)       K++;
        return K;
}
int main()
{
        int Sum = 0;
        int n;
        cin>>n;
        for(int i=1; i<=f1(n); i+=1)
             for(int j=1; j<=i; j++)         Sum++;
        cout<<Sum<<endl;
}
```

25

What is the time complexity of the algorithm:

```
int f1(int n)
{        int K=0;
  for(int j=1; j*j<=n; j++)
    K++;

  return K*K;
}
int main()
{
int Sum = 0;
int n;
cin>>n;
int Terminator = f1(n);
for(int i=1; i<= Terminator; i+=1)
{
  for(int j=1; j<=i; j++)
  {
      Sum++;
  }
}
cout<<Sum<<endl;
}
```

26

What is the time complexity of the algorithm:

```
int f1(int n)
{        int K=0;
         for(int j=0; j*j<=n; j++)
           K++;

        return K;
}
int main()
{
        int Sum = 0;
        int n;
        cin>>n;
        int Terminator = f1(n);
        for(int i=1; i<=Terminator; i+=1)
        {
          for(int j=1; j<=i; j++)
          {
              Sum++;
          }
        }
        cout<<Sum<<endl;
}
```

27

```
for (i=1;i<n;i=i*4)
{
        cout << i;
          for (j=0;j<n;j=j+2)
          {
                  cout << j;
                  sum++
          }
          cout << sum;
}
```

28

```
for (i=1;i<n;i=i*4)
{
        cout << i;
        for (j=0;j<i; j=j+2)
        {
                cout << j;
                sum++
        }
        cout << sum;
}
```

29
for (i=1;i<=n*n;++i)
{
cout << i;
Sum=0;
for (j=1; j<=i; ++j)
{
Sum++;
cout << i;
}
cout << Sum;
}
	30
	for (i=1;i<=n*n*n;++i)
	{
	cout << i;
	Sum=0;
	for (j=1; j<=i; ++j)
	{
	Sum++;
	cout << i;
	}
	cout << Sum;
	}

31
 	
for (i=1;i<=n*n*n; i*=2)
{
cout << i;
Sum=0;
for (j=1;j<=i; j++)
{
Sum++;
cout << i;
}
cout << Sum;
}
	32
	for (i=1;i<=n*n*n; i*=2)
	{
	cout << i;
	Sum=0;
	for (j=1;j<=n; j++)
	{
	Sum++;
	cout << i;
	}
	for (k=1;k<=n; k++)
	{
	Sum++;
	cout << i;
	}
	cout << Sum;
	}

33
for (i=1;i<=n*n*n; i*=2)
{
cout << i;
Sum=0;
for (j=1;j<=i; j++)
{
Sum++;
cout << i;
}
 	
for (j=1;j<=n; j*=2)
{
Sum++;
cout << i;
}
 	
cout << Sum;
}
	34
	for (i=1;i<=n*n*n; i*=2)
	{
	cout << i;
	Sum=0;
	for (j=1;j<=i; j++)
	{
	Sum++;
	cout << i;
	}
	
	for (j=1;j<=n; j++)
	{
	Sum++;
	cout << i;
	}
	
	cout << Sum;
	}

35-36	37

<pre>for (int i=1; i <= n ; i = i * 2)
{
        for (j = 1 ; j <= i ; j = j * 2)
                cout<<"*";
}</pre><pre>for (int i=1; i <= n ; i = i * 2)
        for (j = 1 ; j <= i ; j = j * 2)
                cout<<"*";
 	
for (int i=1; i <= n ; i = i * 2)
        for (j = 1 ; j <= i ; j = j * 2)
                cout<<"*";</pre>
	<pre>for (i=0; i<n; i=i+3)
	{
	        cout << i;
	        for (j=1; j<n; j=j*3)
	        {
	                cout << j;
	                sum++
	        }
	        for (k=1;k<n;k=k*3)
	        {
	                cout << j;
	                sum++
	        }
	
	        cout << sum;
	}</pre>

38	39

<pre>for (int i=1; i <= n ; i = i * 2)
{
        for (j = 1 ; j <= i ; j = j * 2)
        {
                cout<<"*";
        }
}
 	
for(int i=0; i<=N; i++)
{
 Sum++;
}</pre>
	<pre>for (i=0; i<n; i=i+3)
	{
	        cout << i;
	        for (j=1; j<n; j=j*3)
	        {
	                sum++
	        }
	}
	for (k=1;k<n;k=k*3)
	{
	        cout << j;
	        sum++
	}
	cout << sum;</pre>

Question 3 Analyze the complexity Θ of the following functions in terms of N.   1*5+5 = 10 points

<pre>int f1(int N)
{
int Count = 0;
for(int i = 1; i<=N ; i*= 2)
  for(int j=1; j<= i ; j++)
      Count++;
return Count;
}</pre>
	<pre>int f2(int N)
	{
	   int Count=0;
	   int C = f1(N);
	   for(int i=0; i<C; i++)
	       Count++;
	   return Count;
	}</pre>
		<pre>int f5(int N)
		{
		   int Count=0;
		   for(int i=0; i<sqrt(f1(N) * f1(N)); i++)
		       Count++;
		   return Count;
		}</pre>

<pre>int f3(int N)
{
   int Count=0;
   int C = sqrt(f1(N));
   for(int i=1; i<C; i*=2)
       Count++;
   return Count;
}</pre>
	<pre>int f4(int N)
	{
	   int Count=0;
	   for(int i=0; i<f1(N) * f1(N); i++)
	        Count++;
	   return Count;
	}</pre>
		<pre>Int Sum = 0;
		int f6(int N)
		{
		   if(N==1)
		     return 1;
		
		    Sum +=f1(N); Sum +=f2(N);
		    Sum +=f3(N); Sum +=f4(N); Sum +=f5(N);
		   return Sum;
		}</pre>