Q1)

```
for (i=0; i<=n; i++)
    for (j=1; j<=i; j*=2)
        statement block;
```

	1st for	2nd for
n=0	1	0
n=1	2	0+1
n=2	3	0+1+2
n=3	4	0+1+2+2
n=4	5	0+1+2+2+3 - - -

0+1+2+2+3+3+3+3+4.

$$\sum_{k=1}^{n} \left(\lfloor \log_2 k \rfloor + 1 \right)$$

Q2)

Calculate the approximate value of the variable `sum` after the following code fragment, in terms of variable `n`. Use summation notation to compute a closed-form solution (ignore small errors caused by `i` not being evenly divisible by 2). Then use this value to give a tightly bounded Big-Oh analysis of the runtime of the code fragment.

```
int sum = 0;
for (int i = 1; i <= 10; i++) {
    for (int j = n; j <= n + i + 2; j++) {
        sum++;
        sum++;
    }
}

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

$$\left(\sum_{i=1}^{10}\sum_{j=n}^{n+i+2} 2\right) + \sum_{i=1}^{\frac{n}{2}} 1$$

$$\left(\sum_{i=1}^{10} 2(n + i + 2 - n + 1)\right) + \frac{n}{2}$$

$$\left(\sum_{i=1}^{10} 2 \cdot (i + 3)\right) + \frac{n}{2}$$

$$\sum_{i=1}^{10}(2 \cdot i + 6) + \frac{n}{2}$$

$$\sum_{i=1}^{10} 2 \cdot i + \sum_{i=1}^{10} 6 + \frac{n}{2}$$

$$2\sum_{i=1}^{10} i + 60 + \frac{n}{2}$$

$$2\frac{10(11)}{2} + 60 + \frac{n}{2}$$

$$110 + 60 + \frac{n}{2}$$

$$170 + \frac{n}{2}$$

Q3)

[redacted] [redacted]

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

0, 1, 2, 3

n	1st for		2nd for
0	0		0
1	1	x	1
2	4	x	2
3	9	x	3
⋮			
n	n^2	$\times$	n

$n^2 \times n = n^3 = O(n^3)$

Q4)

```
for( i=1; i<=n ; i+=3)
    for( j=1; j<=n; j +=2)
        statement
```

n	1st loop	2nd loop
0	0	0
1	1	1
2	1	2
3	2	2
4	2	3
⋮		
7	2	3
8	2	4
9	3	X 4

$$\left(\lfloor \log_3 n \rfloor + 1\right) \times \left(\lfloor \log_2 n \rfloor + 1\right)$$

Q5)

```
count=0 ; i=1;
do {
    for (j=1; j<=n; j*=2)
        count+=j;
    i++;
}
while (i<=n);
printf("count=%d", count);
```

Big oh $O\,n$

n	while	for
1	1	1
2	2	2
3	3	2
4	4	
	⋮	
n	n	$\times (\lfloor \log_2 n \rfloor + 1)$