Question 1. What is the time complexity for the following code/program?

1.1

```
for (int i = 1; i <= n; i++){
        for (int j = 1; j <= n; j++){
                for (int k = n; k >= 1; k--){
                        sum = i + j + k;
                        cout << sum << endl;}}}
```

let $n = 1$

$n \times m \times m_2 \times constant$

in this case $n \times n \times n \times 1 = n^3$

$\boxed{O(n^3)}$

1.2

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

let $n = 1$

$n \times m \times constant$

in this case $n \times n \times 1 = n^2$

$\boxed{O(n^2)}$

1.3

```
for (int i = 1; i <= n; i++){
        for (int j = 1; j <= 100; j++){
                sum = i + j + k;}}
```

let $n = 1$

Nested but only one bound by n ~~so~~ $n \times 1 + 100 = 100n$

$\boxed{O(n)}$

1.4

```
for (int i = 0; i < n; i++){
        for (int j = i; j > 0; j /= 2){
                cout << j << endl;}}
```

let $n = 1$

First loop (n) | second $j/=2$ ~~so~~ $n/2$ then $n/2^2$ & then $n/2^k$

$n/2^k$ will $=1$ ~~so~~ $\log_2 n = \log_2 2k$ thus

$n \cdot \log_2 n \cdot constant$

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

1.5 A binary search works like this: in a sorted array, the search algorithm compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value and repeating this until the target value is found. If the search ends with the remaining half being empty, the target is not in the array. What is the complexity of binary search? Why?

Immediatly the best case is 1, where we succeed on our first "guess".

Our first array would be n then 2nd would be $n/2$ making it $\frac{n}{2^k}$ which like 1.4 will become 1 & thus $\log_2 n = \log_2 2^k$ giving us the worst out come of $K = \log_2(n)$ or a big $\boxed{O(\log n)}$. This assumes we go all the way down the entirity of the list or the value is actually missing from said list.

Question 2. Finish the following table.

Expression	Dominant term(s)	$O(\ldots)$
$5 + 0.001n^3 + 0.025n$	$0.001n^3$	n^3
$500n + 100n^{1.5} + 50n\log_{10} n$	$100n^{1.5}$	$n^{1.5}$
$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$	$2.5 \cdot n^{1.75}$	$n^{1.75}$
$n^2 \log_2 n + n(\log_2 n)^2$	$n^2 \log_2 n$	$n^2 \log n$
$n \log_3 n + n \log_2 n$	$n \log_2 n$	$n \log n$
$3 \log_8 n + \log_2 \log_2 \log_2 n$	$3 \log_8 n$	$\log n$
$100n + 0.01n^2$	$0.01n^2$	n^2
$0.01n + 100n^2$	$100n^2$	n^2
$2n + n^{0.5} + 0.5n^{1.25}$	$0.5n^{1.25}$	$n^{1.25}$
$0.01n \log_2 n + n(\log_2 n)^2$	$n(\log_2 n)^2$	$n(\log n)^2$
$100n \log_3 n + n^3 + 100n$	n^3	n^3
$0.003 \log_4 n + \log_2 \log_2 n$	$\log_2 (\log_2 n)$	$\log (\log n)$