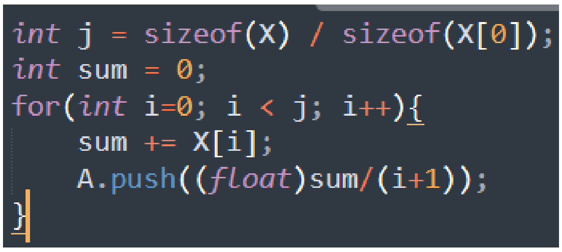
1. 0
2. a. $n^7/5$

   b. ceil(n/2)

   c. k
3. time complexity: O(n)



```
int j = sizeof(X) / sizeof(X[0]);
int sum = 0;
for(int i=0; i < j; i++){
    sum += X[i];
    A.push((float)sum/(i+1));
}
```

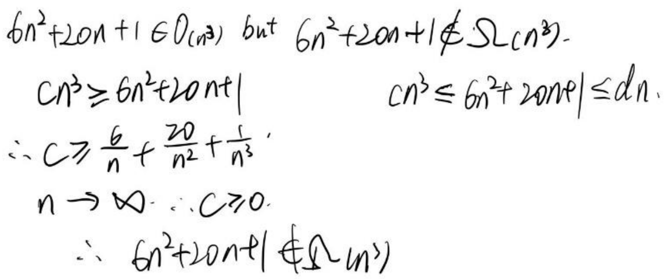
4. (1)The only way for “A” students to figure out their labels is counting all “B” and “C” students. If they find there are already 10 “B” students and 15 “C” students, themselves must be A.

   (2)There are only 10 B and 15 C students, if other students already have B and 15 C students, the left part must be A.
5. (1)First sort the array from smallest to largest, and divide it from the middle of this array into two sub lists. Time complexity is $O(n^2)$.

   (2)
6. (1)i=1, j=1, k=2, m=2, time complexity: O(1).

   (2)i=n, j=n, k=n-1, m=n-1, time complexity: $O(n^4)$.

   (3) First, sort the two-dimensional array from smallest to largest, then compare each two adjacent numbers from smallest to largest, return 1 if they are equal, return 0 if they are not equal.
7.



$6n^2+20n+1 \in O(n^3)$ but $6n^2+20n+1 \notin \Omega(n^3)$.

$cn^3 \geq 6n^2+20n+1$ $\qquad$ $cn^3 \leq 6n^2+20n+1 \leq dn$.

$\therefore c \geqslant \frac{6}{n} + \frac{20}{n^2} + \frac{1}{n^3}$,

$n \to \infty$. $\therefore c \geqslant 0$.

$\therefore 6n^2+20n+1 \notin \Omega(n^3)$

8. Count for the best case: N+2

   Count for the worst case: 1.5N+1
9. Judge from left to right in groups of two plates. If black is on the left and white is on the left, switch their positions, then repeat this step until no such situation occurs.
10. g = O(f) so g<=cf,

    if c=1, so g<=f,

    so f<=f+g<=2f, **f + g = θ(f)**