

2) (10 pts) ANL (Summations and Algorithm Analysis)

Consider the following segment of code, assuming that n has been previously declared and initialized to some positive value:

```
int i, j, k;
for (i = 1; i <= n; i++) {
    for(k = 1; k <= i; k++) {
        j = k;
        while(j > 0)
            j--;
    }
}
```

(a) (3 pts) Write a summation (3 nested sums) equal to the number of times the statement $j--;$ executes, in terms of n .

$$\sum_{i=1}^n \sum_{k=1}^i \sum_{j=1}^k 1$$

Grading: 1/2 pt for outer sum, 1 pt for each inner sum, 1/2 for 1 inside, round down. Note - variable names used in sums are independent of those in code...

(b) (7 pts) Determine a closed form solution for the summation above in terms of n .

$$\begin{aligned} \sum_{i=1}^n \sum_{k=1}^i \sum_{j=1}^k 1 &= \sum_{i=1}^n \sum_{k=1}^i k = \sum_{i=1}^n \frac{i(i+1)}{2} \\ &= \frac{1}{2} \left(\sum_{i=1}^n i^2 + \sum_{i=1}^n i \right) \\ &= \frac{1}{2} \left(\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right) \\ &= \frac{n(n+1)}{12} ((2n+1) + 3) \\ &= \frac{n(n+1)}{12} (2n+4) \\ &= \frac{n(n+1)(n+2)}{6} \end{aligned}$$

Grading: 1 pt solving inner sum, 1 pt solving middle sum, 2 pts for i^2 sum, 1 pt for i sum, 2 pts for algebra - accept either factored or poly form)