Quite a few of you are still struggling with the comparison program. Here I outline the logic of this program that I explained multiple times in class. What's important in this project is the position information of each "letter" (it's called a "base" in DNA sequence) in the alignment block, rather than what the letter is.

I only show the pseudocode important for computing sensitivity and specificity. File input, command line argument processing, many boundary conditions and programming details, and program structure design are skipped.

```
// First process the true alignment file by scanning the true alignment text column by column
    tTotal <-- 0 // total true aligned positions
    T <-- create a large enough array of data type double
    text1 <-- alignment text of the first species
    text2 <-- alignment text of the second species
    pos1 <-- 0 // *** keep track of current position in first species
    pos2 <-- 0 // *** keep track of current position in second species
    for each column col from first to last position of the alignment texts
        if text1[ col ] is a base // i.e., not '-'
            if text2[col] is a base
                T[ pos1 ] <-- pos2  // pos2 of second species is aligned to pos1 of first species
            else
                T[ pos1 ] <-- pos2 - 0.5  // average of pos2-1 and pos2
            increment pos1  // *** important
        if text2[ col ] is a base
            increment pos2  // *** important
    tTotal <-- pos1
    // By this time the array T will be in form of "0   1   2   3   ...  25  25.5 25.5 26  ... "

// Next process the computed alignment file (of maf-format) which may have many alignment blocks
    cTotal <-- 0 // total computed aligned positions
    correct <-- 0 // correctly aligned positions
    For each alignment block
        pos1 <-- start position of the first species  // *** impotant! They are the first integer
        pos2 <-- start position of the second spcies  // *** in an "a line" in the alignment block
        align1 <-- alignment text of the first species
        align2 <-- alignment text of the second species
        for each column col from first to last position of the alignment block
            if ( align1[ col ] is a base
                y <-- T[ pos1 ]  // retrieve true aligned position information from array T
                if ( align2[ col ] is a base
                    y' <-- pos2  // pos2 of second species is aligned to pos1 of first species
                else // '-'
                    y' <-- pos2-0.5
                dif <-- |y-y'|
                if dif <= criteria c
                    increment correct
                increment pos1   // *** important
                increment cTotal
            if ( align2[ col ] is a base
                increment pos2   // *** important

// Finally compute sensitivity and specificity
    sensitivity <-- correct / tTotal
    specificity <-- correct / cTotal
```