```
computeValues(board[][], canVal[], char x, char y, bool promp) //canVal, x, and y are needed for
                                        //isValidNew, and promp is if you want it to be visible
        SET notVal[3][9]                                //what it is not
        IF promp is true                //if from interact not isValidNew
                getCoord(board, x, y)
        SET ix ← uppercase x - 65       //an integer version of x and y
        SET iy ← y - 49

        FOR i = 0; go down column y                     //column in y = 0
                SET notVal[0][i] ←board[i][ix];
        FOR i = 0; go down row x                        //row in y = 1
                SET notVal[1][i] ← board[iy][i];
        SET blockX ← 0
        FOR i = (y/3)*3 run three times                 //block in y = 2
                FOR j = (x/3)*3  run three times        //(x/3)*3 rounds down to nearest 3
                        SET notVal[2][blockX++ ] ← board [i][j]

        SET canX ← 0
        empty(canVal);                                  //empty from previous runs
        FOR n = 1 count to 9
                SET isN ← true                          //if it is a valid number
                SET cn ← '0' + n                        //a char n
                FOR i = 0 count to < 3
                        FOR j = 0 count to < 9
                                IF notVal[i][j] is cn   //if on the not list set false
                                        SET isN ← false
                                FOR k = 0 until canVal[k] is NULL     //avoid duplicates
                                        IF notVal[i][j] is canVal[k]
                                                SET isN ← false
                IF isN is true
                        canVal [ canX++ ] ← cn
        IF promp is true                                //If from interact() not isValidNew()
                PUT The possible values of \'<< y << x << \' are:
                FOR i = 0 until canVal is NULL
                        PUT canVal[i]
                        IF canVal[i+1] is not NULL      //not end
                                PUT comma space
        RETURN
END
```

```
interact(board, fileName)
        SET choice
        SET play ← true

        options()
        display (board)

        WHILE play is true
                PUT newline, carat, space
                GET choice
                SWITCH choice
                        CASE ?
                                options()
                        CASE D
                                display(board)
                        CASE E
                                edit(board)             //I show how it edits in structure chart
                        CASE S
                                SET canVal              //No use here, but used in isValidNew
                                computeValues(board, canVal, 0, 0, true) //0, 0, true means from
                        CASE Q                                          //interact not isValidNew
                                save(fileName, board)                   //in structure chart
                                SET play ← false
                        DEFAULT
                                PUT ERROR: invalid command
        RETURN
END
```