

### ALO (At Least One)

- En este apartado he introducido el código de manera que estaremos asignado como mínimo 1 valor posible para cada celda. Además, imprimiremos al final un 0 para el salto de línea y guardaremos los valores. Ejemplo:

- *alo (1,2,3,4) en la celda 1 del sudoku 2x2*  
1 2 3 4 0
- *alo (5,6,7,8) en la celda 2 del sudoku 2x2*  
5 6 7 8 9 0

```
59 void alo(FILE* f, int *vars, int size) {  
60     /* YOUR CODE HERE */  
61     for(int i=0;i<size;i++) {  
62         fprintf(f,"%d ",vars[i]);  
63     }  
64     fprintf(f,"0\n");  
65 }
```

### AMO (At Most One)

- En este apartado he introducido el código para delimitar que como mucho habrá 1 valor verdadero de entre todos los valores posibles. Ejemplo:

- *amo (1,2,3,4) en la celda 1 del sudoku 2x2*

-1 -2 0

-1 -3 0

-1 -4 0

-2 -3 0

-2 -4 0

-3 -4 0

- *amo (5,6,7,8) en la celda 2 del sudoku 2x2*

-5 -6 0

-5 -7 0

-5 -8 0

-6 -7 0

-6 -8 0

-7 -8 0

```
68 void amo(FILE* f, int *vars, int size) {  
69     /* YOUR CODE HERE */  
70     for(int i=0;i<size-1;i++){  
71         for(int j=i+1;j<size;j++){  
72             fprintf(f,"%d -%d 0\n",vars[i],vars[j]);  
73         }  
74     }  
75 }
```

### EO (Exactly One)

- En este apartado simplemente llamaremos a las funciones anteriormente descritas, el ALO y el AMO.

```
78 void eo(FILE* f, int *vars, int size){  
79     /* YOUR CODE HERE */  
80     alo(f,vars,size);  
81     amo(f,vars,size);  
82 }
```

### Row Constraints

- En este apartado he introducido el código para que se comparen todas las celdas que forman cada fila del sudoku. Finalmente se llama a la función *eo* para que no haya repeticiones. Ejemplo:

- *row constraints (1,5,9,13)* esto es la fila 1 del sudoku 2x2

-1 -5 0  
-1 -9 0  
-1 -13 0  
-5 -9 0  
-5 -13 0  
-9 -13 0

```
135     fprintf(f, "c Row constraints.\n");
136     /* YOUR CODE HERE */
137     for(int i = 0; i < n; i++) {
138         for(int k = 0; k < n; k++) {
139             for(int j = 0; j < n; j++){
140                 vars[j] = x(sudoku, i, j, k);
141             }
142             eo(f, vars, n);
143         }
144     }
```

### Column Constraints

- En este apartado he introducido el código para que se comparen todas las celdas que forman cada columna del sudoku. Finalmente se llama a la función *eo* para que no haya repeticiones. Ejemplo:

- *column constraints (1,17,33,49)* esto es la columna 1 del sudoku 2x2

-1 -17 0  
-1 -33 0  
-1 -49 0  
-17 -33 0  
-17 -49 0  
-33 -49 0

```
147     fprintf(f, "c Column constraints.\n");
148     /* YOUR CODE HERE */
149     for(int j = 0; j < n; j++) {
150         for(int k = 0; k < n; k++) {
151             for(int i = 0; i < n; i++){
152                 vars[i] = x(sudoku, i, j, k);
153             }
154             eo(f, vars, n);
155         }
156     }
```