Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ciencias Básicas

Estructura de Datos y Algoritmos I

Alumno: _Bear Almaraz Miguel Ángel_____

Semestre 2021-2

Nombre de la actividad:

Actividad 03(Miércoles): Sudoku

Fecha: 10/03/2021

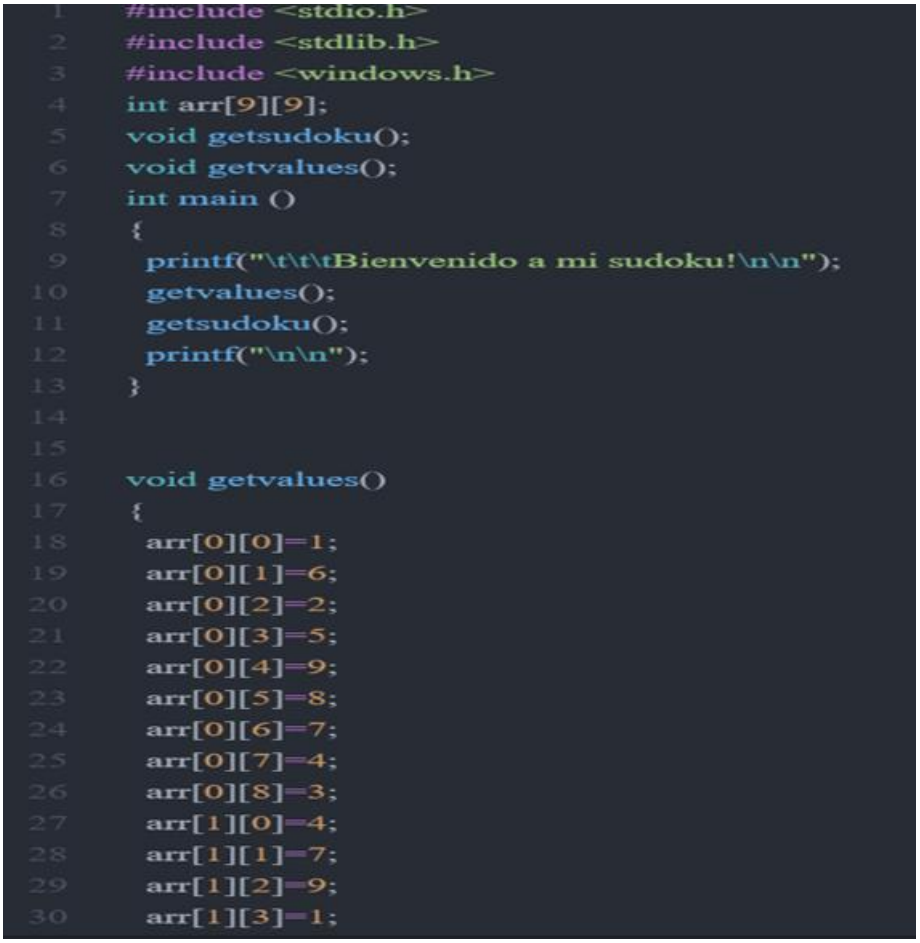
Ejercicio Sudoku

La elaboración de este programa no fue para nada sencillo. Esto debido a que para desarrollar el juego, numerosas cuestiones cuyas soluciones no eran sencillas de encontrar, tuvieron que ser resueltas. Al principio consideré basar la idea principal del juego en la función rand, pero varios problemas surgieron. El principal fue desarrollar un algoritmo que determinara si dentro de una de las nueve matrices se repetía uno de los números (que iban del 0 al 9), y si no era el caso que examinara cada fila y cada columna en busca de un numero repetitivo, y si era el caso cambiar este por otro que entrara dentro del rango ya mencionado.

Es por lo anterior que opte por la opción de resolver un sudoku en línea, registrar los números que eran desplegados en pantalla y los espacios en las matrices que restaban vacíos. Después de haber resuelto el sudoku y haber registrado la información mencionada, procedí a declarar un arreglo bidimensional de 9x9 elementos y a declarar los valores iniciales que registré cuando inicié a resolver el juego en línea.

Después de esto cree una función que pidiera el numero de fila , columna y numero que quería que fuera desplegado en la pantalla, para esto utilice ciclos for y, funciones como printf y scanf.

Debido a que el código es bastante extenso (155 líneas de código), solo adjuntare algunas fotos del contenido del programa y fotos del programa en ejecución.



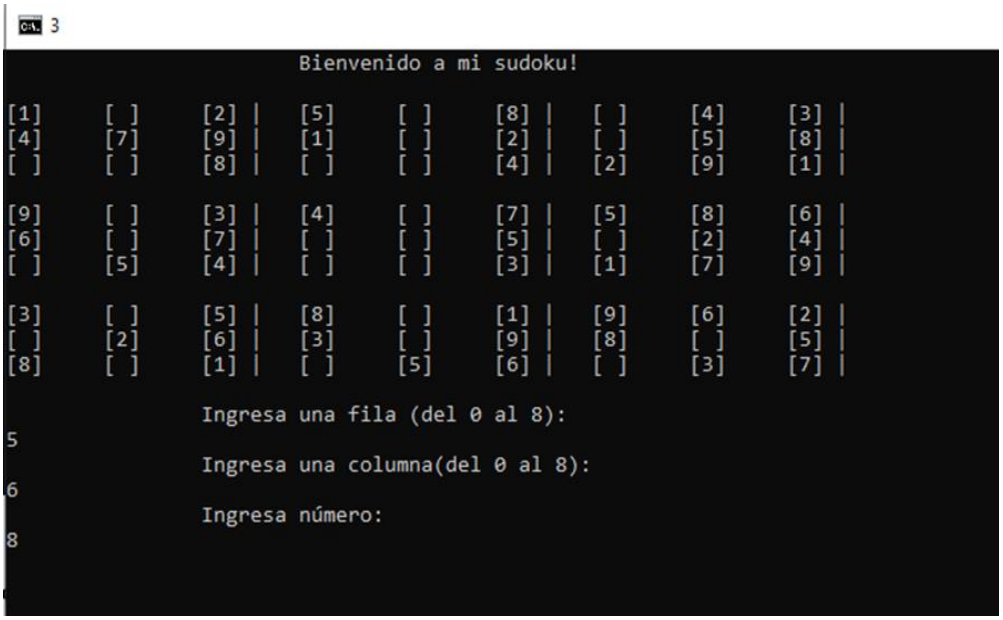
```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int arr[9][9];
void getsudoku();
void getvalues();
int main ()
{
 printf("\t\t\tBienvenido a mi sudoku!\n\n");
 getvalues();
 getsudoku();
 printf("\n\n");
}


void getvalues()
{
 arr[0][0]=1;
 arr[0][1]=6;
 arr[0][2]=2;
 arr[0][3]=5;
 arr[0][4]=9;
 arr[0][5]=8;
 arr[0][6]=7;
 arr[0][7]=4;
 arr[0][8]=3;
 arr[1][0]=4;
 arr[1][1]=7;
 arr[1][2]=9;
 arr[1][3]=1;
```

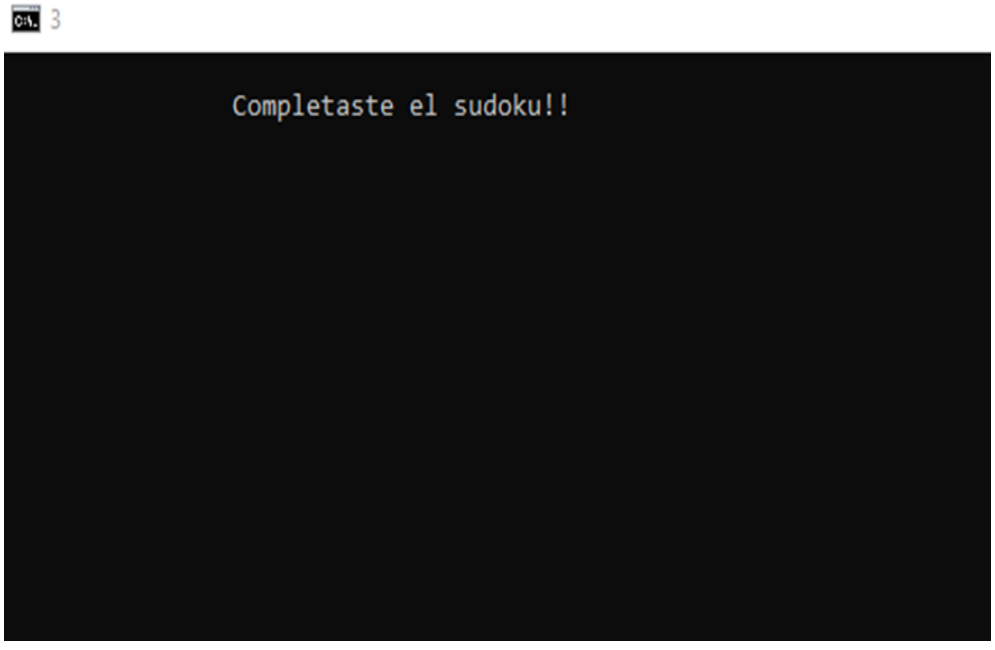
```
30          arr[1][3]=1;
31          arr[1][4]=3;
32          arr[1][5]=2;
33          arr[1][6]=6;
34          arr[1][7]=5;
35          arr[1][8]=8;
36          arr[2][0]=5;
37          arr[2][1]=3;
38          arr[2][2]=8;
39          arr[2][3]=7;
40          arr[2][4]=6;
41          arr[2][5]=4;
42          arr[2][6]=2;
43          arr[2][7]=9;
44          arr[2][8]=1;
45          arr[3][0]=9;
46          arr[3][1]=1;
47          arr[3][2]=3;
48          arr[3][3]=4;
49          arr[3][4]=2;
50          arr[3][5]=7;
51          arr[3][6]=5;
52          arr[3][7]=8;
53          arr[3][8]=6;
54          arr[4][0]=6;
55          arr[4][1]=8;
56          arr[4][2]=7;
57          arr[4][3]=9;
58          arr[4][4]=1;
59          arr[4][5]=5;
```

```
100  void getsudoku()
101  {
102   int i, j, r, c, num, f, w;
103   f=0;
104   w=0;
105   do {
106    for(i=0; i<9; i++)
107    {
108     if(i==3 || i==6)
109     {
110      printf("\n");
111     }
112     for(j=0; j<9;j++)
113     {
114      if(j==2 || j==5 || j==8)
115      {
116       printf("[%d] |\t", arr[i][j]);
117      }
118      else if((i==0 && j==1)||(i==0 && j==2)||(i==0 && j==4)||(i==0 && j==5)||
119      (i==0 && j==6)||(i==1 && j==2)||(i==1 && j==4)||(i==1 && j==6)
120      ||(i==2 && j==0)||(i==2 && j==1)||(i==2 && j==2)||(i==2 && j==3)||
121      (i==2 && j==4)||(i==2 && j==5)||(i==3 && j==1)||
122      (i==3 && j==4)||(i==3 && j==5)||(i==4 && j==1)||(i==4 && j==2)||
123      (i==4 && j==3)||(i==4 && j==4)||(i==4 && j==6)||
124      (i==5 && j==0)||(i==5 && j==2)||(i==5 && j==3)||(i==5 && j==4)||
125      (i==5 && j==5)||(i==5 && j==8)||(i==6 && j==1)||(i==6 && j==4)||
126      (i==6 && j==5)||(i==7 && j==0)||(i==7 && j==4)||(i==7 && j==7)||
127      (i==8 && j==1)||(i==8 && j==2)||(i==8 && j==3)||(i==8 && j==5)||
128      (i==8 && j==6))//end of if parentheses
129      {//start of if
```

```
        (i==6 && j==5)||(i==7 && j==0)||(i==7 && j==4)||(i==7 && j==7)||
        (i==8 && j==1)||(i==8 && j==2)||(i==8 && j==3)||(i==8 && j==5)||
        (i==8 && j==6))//end of if parentheses
        {//start of if
          printf("[ ]\t");
        }//end of if*/
        else
        {
           printf("[%d]\t", arr[i][j]);
        }
        if(j==8)
        {
         printf("\n");
        }
       }//end j for loop
      }//end i for loop
       printf("\n\t\tIngresa una fila (del 0 al 8):\n");
       scanf("%d", &r);
       printf("\t\tIngresa una columna(del 0 al 8):\n");
       scanf("%d", &c);
       printf("\t\tIngresa n%cmero:\n", 163);
       scanf("%d", &num);
       system("cls");
      f=1;
    } while(f=0);
    printf("\n\t\tCompletaste el sudoku!!\n");
    Sleep(5000);
```

```
C:\. 3

                    Bienvenido a mi sudoku!

[1]     [ ]     [2] |   [5]     [ ]     [8] |   [ ]     [4]     [3] |
[4]     [7]     [9] |   [1]     [ ]     [2] |   [ ]     [5]     [8] |
[ ]     [ ]     [8] |   [ ]     [ ]     [4] |   [2]     [9]     [1] |

[9]     [ ]     [3] |   [4]     [ ]     [7] |   [5]     [8]     [6] |
[6]     [ ]     [7] |   [ ]     [ ]     [5] |   [ ]     [2]     [4] |
[ ]     [5]     [4] |   [ ]     [ ]     [3] |   [1]     [7]     [9] |

[3]     [ ]     [5] |   [8]     [ ]     [1] |   [9]     [6]     [2] |
[ ]     [2]     [6] |   [3]     [ ]     [9] |   [8]     [ ]     [5] |
[8]     [ ]     [1] |   [ ]     [5]     [6] |   [ ]     [3]     [7] |

            Ingresa una fila (del 0 al 8):
```


3
Bienvenido a mi sudoku!
[1] [ ] [2] | [5] [ ] [8] | [ ] [4] [3] |
[4] [7] [9] | [1] [ ] [2] | [ ] [5] [8] |
[ ] [ ] [8] | [ ] [ ] [4] | [2] [9] [1] |
[9] [ ] [3] | [4] [ ] [7] | [5] [8] [6] |
[6] [ ] [7] | [ ] [ ] [5] | [ ] [2] [4] |
[ ] [5] [4] | [ ] [ ] [3] | [1] [7] [9] |
[3] [ ] [5] | [8] [ ] [1] | [9] [6] [2] |
[ ] [2] [6] | [3] [ ] [9] | [8] [ ] [5] |
[8] [ ] [1] | [ ] [5] [6] | [ ] [3] [7] |
Ingresa una fila (del 0 al 8):
5
Ingresa una columna(del 0 al 8):
6
Ingresa número:
8


3
Completaste el sudoku!!