



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Estructura de Datos y Algoritmos I

Actividad Asíncrona #03 Miércoles

Sudoku en C

Alumno:

Flores Ramírez Eduardo Amilcar



10 de marzo del 2021

Sudoku en C

Un arreglo comprende una secuencia finita y ordenada de posiciones de memoria que almacena datos del mismo tipo, organizados en una misma columna y en uno o más renglones. De acuerdo con los tipos de arreglos estudiaremos dos, los cuales comprenden los arreglos unidimensionales y los arreglos bidimensionales.

Arreglo unidimensional: Un arreglo Unidimensional, es considerado el tipo más simple, comprende una sola dimensión y está conformado por n número de elementos del mismo tipo.

Arreglo bidimensional: Los arreglos Bidimensionales también conocidos como tablas o matrices, comprende una composición de filas y columnas de elementos del mismo tipo.

Desarrollo:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int sudoku[9][9]=
5  {{8,0,0,0,0,0,6,0,0},
6   {0,2,9,6,7,0,0,1,0},
7   {0,0,0,0,1,4,0,5,0},
8   {6,0,0,3,9,1,5,0,2},
9   {0,5,1,0,0,0,9,0,0},
10  {9,0,2,0,0,6,0,0,0},
11  {0,6,0,4,3,0,0,0,0},
12  {0,9,0,0,8,7,1,6,0},
13  {0,0,7,0,0,0,0,0,3}};
14
15  int tablero[9][9];
16
17  int cont=0;
18
19  void imprimir_sudoku()
20  {
21      int x,y;
22      for(x=0;x<9;x++)
23      {
24          for(y=0;y<9;y++)
25          {
26              printf("%d", sudoku[y]);
27              if(y==2 || y==5)
28                  printf("|");
29          }
30          printf("\n");
31          if(x==2 || x==5)
32          {
33              printf("---+---+---");
34              printf("\n");
35          }
36      }
37      printf("\n");
38  }
39
40  void inicializar(int x, int y)
41  {
42      int a,b;
43      a=x;b=y+1;
44      while(a<9)
45      {
46          if(tablero[a]==0) {sudoku[a]=0;}
47          b++;
48          if (b>8) { b=0;a++; }
49      }
50  }

```

```

51
52 int comprobar(int x, int y)
53 {
54     int a,b,c,correcto,retorno,contador,x2,y2;
55     correcto=1;
56     for(a=1;a<=9 && correcto;a++)
57     {
58         contador=0;
59         for(b=0;b<9 && correcto;b++)
60         {
61             if(sudoku!=0)
62                 if(sudoku==a)
63                     contador++;
64         }
65         if (contador>1) correcto=0;
66     }
67     retorno=correcto;
68     if(correcto)
69     {
70         for(a=1;a<=9 && correcto;a++)
71         {
72             contador=0;
73             for(b=0;b<9 && correcto;b++)
74             {
75                 if(sudoku[y]!=0)
76                     if(sudoku[y]==a)
77                         contador++;
78             }
79             if(contador>1) correcto=0;
80         }
81         retorno=correcto;
82     }
83
84     if(correcto)
85     {
86         for(a=1;a<=9 && correcto;a++)
87         {
88             x2=(x-(x % 3));
89             y2=(y-(y % 3));
90             contador=0;
91             for(b=x2;b<x2+3;b++)
92             {
93                 for(c=y2;c<y2+3;c++)
94                 {
95                     if(sudoku[c]!=0)
96                         if(sudoku[c]==a)
97                             contador++;
98                 }
99             }
100             if(contador>1) correcto=0;

```

```

100         if(contador>1) correcto=0;
101     }
102     retorno=correcto;
103 }
104 return retorno;
105 }
106
107 int resolver_sudoku(int x, int y)
108 {
109     int a, correcto;
110
111     if (y>8) (y=0;x++);
112     while(x<9 && tablero[y]!=0)
113     {
114         y++;
115         if (y>8) { y=0;x++; }
116     }
117     if(x==9)
118     {
119         if(!comprobar(8, 8))
120         {
121             printf("Sin solucion");
122             exit(-1);
123         } else
124         {
125             imprimir_sudoku();
126             cont++;
127             if(cont==100)
128                 exit(-1);
129         }
130     } else
131     {
132         for(a=1;a<=9;a++)
133         {
134             inicializar(x,y);
135             sudoku[y]=a;
136             correcto=comprobar(x,y);
137             if(correcto)
138                 resolver_sudoku(x,y+1);
139             else{
140                 sudoku[y]=0;
141             }
142         }
143     }
144 }
145
146 int main()
147 {
148     int x,y,correcto;
149     for(x=0;x<9;x++)
150     for(y=0;y<9;y++)

```

```

151     tablero[y]=sudoku[y];
152
153     correcto=1;
154     for(x=0;x<9 && correcto;x++)
155     for(y=0;y<9 && correcto;y++)
156     {
157         correcto=comprobar(x,y);
158     }
159
160
161     if(!correcto)
162     printf("Datos de entrada incorrectos\n");
163     else
164     {
165         resolver_sudoku(0,0);
166     }
167
168     printf("Combinaciones : %d\n", cont);
169 }
170

```

*El Pseudocódigo no funcionó como se esperaba, no pude pasar a la parte de consola debido a que el programa no me dejaba correr el código debido a un error en el renglón 46, en donde la asignación está asociada a una matriz.

```

46      error: assignment to expression with array type

```

El objetivo principal considero que si se cumplió ya que conocí y pude utilizar y declarar arreglos unidimensionales como también bidimensionales, los cuales sirvieron en la práctica para poder generar área en donde trabajaríamos con los valores del Sudoku, pero considero que no pude llegar a el objetivo final debido a que mis conocimientos en antecedentes son muy escasos, y aun no entiendo muchos de los comandos que aquí se presentan.

Referencias.

elearningcorporativo. (2016, 20 abril). Arreglos Unidimensionales y Bidimensionales.

Recuperado 13 de marzo de 2021, de

<https://elearningcorporativo.wordpress.com/2015/08/18/hola-mundo/>