

## Trabalho de Programação

### Atividade 4 - Sudoku

## Objetivo

Utilizar matrizes e funções para resolver um problema de programação.

## Problema

"Sudoku<sup>1</sup>, por vezes escrito Su Doku é um jogo baseado na colocação lógica de números. O objetivo do jogo é a colocação de números de 1 a 9 em cada uma das células vazias numa grade de 9x9, constituída por 3x3 subgrades chamadas regiões. O quebra-cabeça contém algumas pistas iniciais, que são números inseridos em algumas células, de maneira a permitir uma indução ou dedução dos números em células que estejam vazias. Cada coluna, linha e região só pode ter um número de cada um dos 1 a 9. Resolver o problema requer apenas raciocínio lógico e algum tempo. Os problemas são normalmente classificados em relação à sua realização [...]. Foi criado por Howard Garns, um projetista e arquiteto de 74 anos aposentado".

O seu trabalho é criar um programa que leia um valor  $n$ ,  $n = \{2, 3, 4\}$ , que determina a dimensão de uma matriz quadrada  $n^2 \times n^2$  e em seguida leia  $n^4$  valores que preenchem uma matriz quadrada que representa um jogo de sudoku resolvido. A matriz pode ser 4 x 4 (se  $n = 2$ ), 9 x 9 (se  $n = 3$ ) ou 16 x 16 (se  $n = 4$ ). Com a matriz carregada na memória, o seu programa deve avaliar se o preenchimento está correto ou não, indicando a primeira linha e coluna onde foi encontrado o erro (se houver), ou a mensagem "Sudoku Ok!". Para o jogo de Sudoku 16 x 16, considere as letras A, B, C, D, E, F e G para representar os dígitos 10, 11, 12, 13, 14, 15 e 16, respectivamente.

Para entrada:

```
3
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9
```

O programa deverá apresentar o seguinte resultado:

```
+-----+-----+-----+
| 5  3  4 | 6  7  8 | 9  1  2 |
| 6  7  2 | 1  9  5 | 3  4  8 |
```

<sup>1</sup>Sudoku (wikipedia): <https://pt.wikipedia.org/wiki/Sudoku>

```

| 1  9  8 | 3  4  2 | 5  6  7 |
+-----+-----+-----+
| 8  5  9 | 7  6  1 | 4  2  3 |
| 4  2  6 | 8  5  3 | 7  9  1 |
| 7  1  3 | 9  2  4 | 8  5  6 |
+-----+-----+-----+
| 9  6  1 | 5  3  7 | 2  8  4 |
| 2  8  7 | 4  1  9 | 6  3  5 |
| 3  4  5 | 2  8  6 | 1  7  9 |
+-----+-----+-----+
Sudoku Ok!

```

## Descrição

1. Desenvolver o programa conforme especificado com o nome atividade4.c.
2. Completar a função sudokuOk, conforme os comentário no código fonte.
3. Testar o programa para vários valores de entrada.

```

$./atividade4 < jogo1.txt

```

---

```

1  #include <stdio.h>
2
3  void leSudoku(int n, int sudoku[16][16])
4  {
5      char s[2];
6      int i, j, dim = n * n;
7      for (i = 0; i < dim; i++)
8          for (j = 0; j < dim; j++)
9              {
10                 scanf("%s", s);
11                 if (s[0] >= '1' && s[0] <= '9')
12                     sudoku[i][j] = s[0] - '0';
13                 if (s[0] >= 'A' && s[0] <= 'G')
14                     sudoku[i][j] = s[0] - 'A' + 10;
15             }
16 }
17
18 void imprimeLinha(int n)
19 {
20     int i, j;
21     for (i = 0; i < n; i++)
22     {
23         printf("+");
24         for (j = 0; j < n; j++)
25             printf("---");
26     }
27     printf("+\n");
28 }
29
30 void imprimeSudoku(int n, int sudoku[16][16])
31 {
32     int i, j, dim = n * n;
33     imprimeLinha(n);
34     for (i = 0; i < dim; i++)

```

```

35 {
36     printf("|");
37     for (j = 0; j < dim; j++)
38     {
39         int num = sudoku[i][j];
40         if (num <= 9)
41             printf("%2d ", num);
42         else
43             printf("%2c ", num - 10 + 'A');
44         if ((j + 1) % n == 0)
45             printf("|");
46     }
47     printf("\n");
48     if ((i + 1) % n == 0)
49         imprimeLinha(n);
50 }
51 }
52
53 /*-----
54 * Funcao que testa uma solucao da Sudoku
55 * Parametros:
56 *     n = dimensao das regioes
57 *     sudoku - jogo preenchido
58 *     lin - linha do erro (se houver)
59 *     col - coluna do erro (se houver)
60 * Retorna:
61 *     1 se a matriz e uma solucao do sudoku
62 *     0 caso contrario, modificando os parametros
63 *     lin e col para indicar a linha e coluna onde
64 *     foi detectado o problema.
65 *-----*/
66 int sudokuOk(int n, int sudoku[16][16], int *lin, int *col)
67 {
68     // TO DO
69     return 1;
70 }
71
72 int main(void)
73 {
74     int n, nlin, ncol, ok;
75     int sudoku[16][16];
76     scanf("%d", &n);
77     leSudoku(n, sudoku);
78     imprimeSudoku(n, sudoku);
79     ok = sudokuOk(n, sudoku, &nlin, &ncol);
80     if (ok)
81         puts("Sudoku Ok!");
82     else
83         printf("Erro na posicao (%d,%d)\n", nlin, ncol);
84     return 0;
85 }

```

---

## Entrega

1. Incluir um comentário no cabeçalho de cada programa fonte com o seguinte formato:

```

1 /*-----
2 *          UNIFAL – Universidade Federal de Alfenas.
3 *          BACHARELADO EM CIENCIA DA COMPUTACAO.

```

4 \* Trabalho...: Atividade 4  
5 \* Disciplina: Programacao  
6 \* Professor.: Luiz Eduardo da Silva  
7 \* Aluno.....: Fulano da Silva  
8 \* Data.....: 99/99/9999  
9 \*

---

2. Enviar num arquivo único (.ZIP), com o(s) arquivo(s) fonte(s) do projeto através do Envio de Arquivo do MOODLE.