

UNIVERSIDADE DE BRASÍLIA – UnB
FACULDADE DO GAMA – FGA

Prova Prática– 2017/1

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES
PROVA PRÁTICA 3

Prof. Matheus Vieira Portela

Objetivo Essa etapa de avaliação visa que o aluno aprenda a utilizar **funções, structs, vetores e matrizes** em programas de computador que resolvam problemas reais.

Contextualização O *Sudoku*¹ é um quebra-cabeças que foi criado nos anos 70 pelo arquiteto Howard Garns. Formado por um *grid* com 9 linhas e 9 colunas, o objetivo do jogador é preenchê-lo com números de 1 a 9 de forma que não haja repetição de números na mesma linha, coluna ou região (quadrados menores formados por 3x3 células).

Inicialmente, o Sudoku possui algumas pistas: células com valores pré-definidos. Dessa forma, cabe ao jogador preencher as células vazias com dígitos sem que haja conflitos com os números iniciais².

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

Figura 1: Jogo de Sudoku, com as pistas iniciais.

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

Figura 2: Jogo de Sudoku resolvido, com os dígitos definidos pelo jogador em vermelho.

Problema Escreva um programa de computador que leia um problema de *Sudoku* do usuário e escreva na tela a sua resolução.

¹Veja mais em <https://pt.wikipedia.org/wiki/Sudoku>.

²Caso você nunca tenha jogado *Sudoku*, assista ao tutorial com o passo a passo em <https://www.youtube.com/watch?v=t3nx8axVx1k>.

Entrada O programa deve ler o valor de todas as 81 células do jogo de *Sudoku* do terminal conforme o exemplo abaixo. Células vazias serão marcadas com o caracter 'X'.

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

Saída O programa deve escrever na tela a solução do *Sudoku* conforme o exemplo abaixo: escreva cada número da região, separando-os por espaço, e utilize os caracteres '|' e '-' para separar as regiões. Não deve haver espaços após o último dígito de cada linha.

Atente-se à formatação exigida. Programas com formatação incorreta serão considerados incorretos, mesmo se a diferença for apenas um caracter de espaço.

```
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
```

Critérios de Avaliação Todos os programas serão avaliados de duas formas: correção automática do Moodle, para avaliar a corretude da solução apresentada, e avaliação do código, de acordo com os critérios a seguir. Programas que não estiverem corretos de acordo com a correção automática **receberão nota zero automaticamente**.

Para os programas que estiverem corretos de acordo com a correção automática do Moodle, a nota será proporcional aos seguintes critérios de avaliação:

1. Correção automática do Moodle: 50%
2. Utilização de funções: 20%
3. Utilização de **structs**: 10%
4. Nomes mnemônicos para variáveis: 10%

Programas que forem identificados como **plágio receberão nota zero**.

Observações Gerais O programa deve ser entregue pelo Moodle até **30/06/2017 (sexta-feira), às 23h55**. Sob hipótese alguma serão aceitas as resoluções após prazo de entrega, independentemente de fatores inesperados, como queda de energia ou sobrecarga nos servidores do Moodle. *Planejem-se com antecedência!*

O programa deve ser realizado **individualmente ou em dupla**, entretanto, não devem ser feitos em grupo. Duplas com alto grau de similaridade entre suas soluções receberão ambas **nota zero**. *Sugestão:* para evitar tal situação, discuta métodos de solução do problema sem permitir que outros alunos vejam seu programa e nunca escreva código em C diretamente para outras duplas.

Para fins de identificação, **inclua nome, matrícula e turma de cada aluno no topo do programa na forma de comentário** (entre `/* */` ou `//`), conforme o exemplo a seguir. Programas que não estiverem devidamente identificados receberão **nota zero**.

```
1  /*
2  Autores:
3  Hermione Jean Granger - 16/0012345 - JJ
4  Ginevra Molly Weasley - 17/0067890 - GG
5  */
6
7  #include <stdio.h>
8
9  ...
```