

### TAD: SUDOKU

**Data da Entrega:** 23/12/12

**Valor:** 1 ponto

+0.1 extra documentação Doxygen em HTML

+0.2 extra análise empírica completa com gráficos

**Objetivos:**

- Implementar um tipo abstrato de dados.
- Calcular complexidade de operações simples.
- Virar um ninja na verificação de um Sudoku.

**Conteúdo a ser entregue:**

- Código fonte e executável devem ser depositados no Moodle.
- Relatório deve ser depositado no Moodle.
- Sempre cite suas fontes: pessoas, livros, sítios.

### O JOGO SUDOKU

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

Neste trabalho prático você deverá implementar um verificador de soluções para o jogo Sudoku e também um aconselhador de valores válidos<sup>1</sup>. No Sudoku, um tabuleiro vem parcialmente preenchido e você deve encontrar números que o completem, mantendo linhas, colunas e regiões válidas: sem números repetidos. Uma célula específica do jogo será referenciada pela sua linha e coluna. Por exemplo, a célula destacada em azul é a (3,6). A região em verde é a 6, e para que a solução seja válida as células vazias não podem ser preenchidas com 1 e 7. Para que a linha 2 (em

<sup>1</sup> Se você nunca jogou Sudoku (o que eu duvido), então leia sobre o assunto na rede. Há vários sítios explicando as regras e inclusive permitindo que você jogue online (por exemplo, <http://www.sudoku.name/index-pt.php>).

rosa) seja válida, as células vazias só poderão ser preenchidas com 2,6,7 ou 8. Já na coluna 3 (amarela), só podemos colocar 1,2,5 ou 9.

## ARQUIVOS DE ENTRADA

Os tabuleiros serão informados para seu programa através de arquivos. O nome do arquivo deverá ser informado na linha de comando e seguir o formato (entradas vazias são representadas por 0) onde há uma linha do arquivo por linha do tabuleiro. O tabuleiro acima seria representado como:

sud1.txt

```
0 1 4 5 2 0 0 0 9
3 0 0 0 9 0 4 5 1
5 0 8 0 0 0 0 6 7
0 0 0 0 0 3 7 0 0
0 0 6 2 0 5 1 0 0
0 0 3 4 0 0 0 0 0
4 3 0 0 0 0 6 0 2
8 2 7 0 4 0 0 0 5
1 0 0 0 5 2 8 4 0
```

## SAÍDA DO SEU PROGRAMA

Seu objetivo é construir um programa para verificar se um preenchimento (parcial) de um tabuleiro é válido e, caso afirmativo, para cada célula vazia, indicar quais são os valores válidos para mesma. Se o tabuleiro é inválido, todas as consistências deverão ser apontadas. O tabuleiro será lido de um arquivo de entrada, fornecido como parâmetro:

```
.\sudoku <arquivo_entrada>
```

A saída dependerá do estado do tabuleiro e deverá seguir exatamente os formatos sugeridos abaixo.

1. O tabuleiro completo (todas as células preenchidas) e válido (sem valor duplicado na linha, coluna ou região), só informe.

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

Jogo completo. Voce ganhou!

2. O tabuleiro incompleto e válido: informe os valores possíveis para as células vazias.

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

Voce esta no caminho certo. Sugestoes:

(3,4): 1 4

(3,6): 1 4

(6,4): 1 4

(6,5): 1 4

(7,6): 1

(8,5): 4

3. O tabuleiro inválido (completo ou não): você deverá indicar TODAS as inconsistências.

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

Alguma coisa deu errado... Invalidos:

Linha 2: (2,1) e (2,8)

Linha 4: (4,1) e (4,2)

Coluna 1: (3,1) e (4,1)

Coluna 8: (2,8) e (5,8)

Regiao 3: (1,7) e (2,8)

Regiao 4: (4,1) e (4,2)

## IMPLEMENTAÇÃO

Para atingir seu objetivo, você deverá construir um Tipo Abstrato de Dados `Tabuleiro` como representação do Sudoku que você quer resolver. O TAD deverá implementar, pelo menos, as seguintes operações:

1. `Tabuleiro inicializa(NomeArquivo)`: inicializa tabuleiro a partir de arquivo
2. `Celula[] defineVazias(Tabuleiro)`: retorna célula vazias do tabuleiro
3. `boolean éValido(Tabuleiro)`: verifica se uma célula é válida
4. `Inteiro[] valoresValidos(Tabuleiro,Celula)`: retorna todos os valores válidos para uma célula vazia

onde `Celula` é o par (linha,coluna). O TAD deve ser implementado utilizando a separação interface no `.h` e implementação `.c` discutida em sala, bem como as convenções de tradução. Caso a operação possa dar errado, devem ser definidos retornos com erro, tratados no corpo principal.

O código do arquivo `main.c` **somente** poderá ser alterado nos lugares indicados ou para declaração de variáveis, ou seja, a assinatura das funções está definida pelo uso.

Se necessário, você poderá definir e implementar outras operações para seu TAD. A implementação da Estrutura de Dados do TAD `Tabuleiro` deverá necessariamente utilizar um arranjo de duas dimensões para representar ou o tabuleiro completo ou as regiões, você escolhe.

## A SER ENTREGUE

1. Código fonte comentado e executável (compilado para Windows) - Moodle
  - a. É mandatório seguir a divisão em `.h` e `.c` e seguir as regras de implementação de TAD.
  - b. É mandatório a utilização de pelo menos uma estrutura bidimensional.
  - c. Indentação, comentários e clareza são essenciais para boa compreensão do código.
  - d. Cópia e plágio serão punidos com ZERO.
  - e. Ponto extra para quem documentar usando Doxygen.
2. Relatório com a seguinte estrutura: - Moodle e impresso.
  - a. Introdução: descrição geral do problema.
  - b. Tipo Abstrato de Dados: Descrição da TAD e da estrutura de dados utilizada para implementá-la.
    - i. Discussão do impacto de se alterar o tamanho do grid para outro valor (4 ou 16, por exemplo) na sua implementação.
  - c. Análise de complexidade
    - i. Teórica: para as quatro funções descritas acima, em função do tamanho do grid  $n$  (no nosso caso, 9).
    - ii. Empírica: medida de tempo para os exemplos disponibilizados no Moodle. Descrever com tempo foi medido e apresentar gráficos.
3. Conclusões: o que você achou do trabalho, o que teve mais dificuldade.

## AVALIAÇÃO

	1.3
Código	0.6
Relatório	0.4
Extras	0.3