

## Trabalho Prático

### 1 Descrição do Trabalho

Você será responsável por programar um jogo Sudoku em C++, utilizando dos conceitos vistos na disciplina. Mas este não é um Sudoku qualquer; um novo Sudoku deve ser carregado de um arquivo texto e todo jogo será salvo em um arquivo binário, permitindo que possa ser recuperado posteriormente.

### 2 Sudoku

Sudoku é um quebra-cabeça que explodiu mundialmente em popularidade em 2005. Quase todo grande jornal agora publica um jogo de Sudoku diariamente. Um Sudoku completo é um quadro 9x9 (ou seja, um arranjo de duas dimensões) com dígitos entre 1 e 9, conforme apresentado na Figura 1.

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

Figura 1: Exemplo de um Sudoku resolvido.

Em um típico desafio Sudoku, é fornecido um quadro com alguns campos preenchidos e outros não. A tarefa do jogador consiste em preencher os campos vazios de forma a completar o quebra-cabeça. Alguns são fáceis de resolver; outros são bem difíceis, requerindo o emprego de estratégias sofisticadas (como *backtracking*). A Figura 2 apresenta um Sudoku inicial.

### 3 Regras

As regras do Sudoku são bem simples:

1. Não podem existir números repetidos na linha;

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

Figura 2: Exemplo de um Sudoku inicial.

2. Não podem existir números repetidos na coluna;
3. Não podem existir números repetidos no mesmo quadrante. Um quadrante corresponde a um conjunto 3x3. A Figura 3 apresenta os quadrantes com os centros numerados.

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

Figura 3: Representação dos quadrantes do Sudoku.

Neste trabalho, iremos considerar somente o Sudoku de nível fácil. Isto implica que, das posições vazias restantes, sempre existirá uma posição em que só é possível um valor, ou seja, todos os outros valores violariam uma das regras apresentadas anteriormente. Por exemplo, dado o Sudoku da Figura 2, na posição (0, 3) o único valor válido possível para esta posição é o valor 3 (tente você mesmo verificar isto).

## 4 Funcionamento Básico do Programa

A estrutura do código já está montada no arquivo fornecido `sudoku.cpp`. Todas as funções necessárias para o funcionamento correto já estão declaradas. Não é necessário a adição de nenhuma nova função. Você deve implementar todas as funções que estão marcadas como

TODO. Para auxiliá-lo no que cada função deve realizar, toda função possui além do seu nome descritivo, um breve comentário do que a mesma faz. Baseado nisso você deve implementar a lógica.

É interessante primeiramente que você invista tempo para compreender o código já existente e o que falta, pois grande parte da sua implementação deve ser feita invocando funções já implementadas. O funcionamento do programa segue abaixo:

```
~~~~~ SUDOKU ~~~~~
[1] - Carregar jogo"
[2] - Preencher um campo"
[3] - Resolver um passo"
[4] - Resolver"
[9] - Finalizar"
```

-----

Opcao:

Ao selecionar a opção 1 no menu principal você irá para o menu abaixo.

```
~~~~~ MENU ARQUIVO ~~~~~
[1] - Carregar novo Sudoku
[2] - Continuar jogo
[9] - Voltar ao menu anterior
```

-----

Opcao:

Neste menu é apresentado as opções 1) Carregar novo Sudoku e 2) Continuar jogo. Na opção 1 será solicitado o nome de um arquivo `.txt` para carregar o Sudoku. Um exemplo de entrada segue a seguir:

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

Figura 4: Exemplo de entrada.

O formato aceito é um conjunto de 9 linhas por 9 colunas, com cada posição contendo um número entre 0 a 9. O valor 0 é um número especial, indicando que aquela posição está vazia.

A opção 2 é utilizada para continuar um jogo Sudoku. É solicitado o nome do arquivo (binário) que você deseja continuar. Toda vez que você inicia um novo jogo é criado um arquivo binário de nome aleatório para armazenar este jogo. Esse arquivo binário possui o seguinte formato: um número inteiro,  $n$ , informando o número de jogadas e  $n + 1$  quadros. Sempre que uma jogada é feita e essa jogada for válida, o quadro é salvo no arquivo binário. Existe um quadro a mais que é salvo pois assim que um novo jogo é criado, cria-se o arquivo e então já é salvo o estado inicial do quadro.

```
~~~~~ SUDOKU ~~~~~"
[1] - Carregar jogo"
[2] - Preencher um campo"
[3] - Resolver um passo"
[4] - Resolver"
[9] - Finalizar"
```

-----  
Opcao:

Voltando ao menu principal, o restante das opções é descrito abaixo:

**Opção 2 - Preencher um campo** solicita-se a linha, a coluna e o valor. Antes de simplesmente inserir o valor na posição fornecida, é verificado se o valor fornecido é válido, ou seja, se segue as regras do Sudoku;

**Opção 3 - Resolver um passo** esta opção resolve um passo do Sudoku, ou seja, preenche um campo com o valor correto;

**Opção 4 - Resolver** resolve o Sudoku inteiro;

**Opção 9 - Finalizar** termina o programa;

Considere que não ocorra entrada incorreta dos dados. Por exemplo, se é solicitado um número inteiro do usuário, espere que seja fornecido um número inteiro.

## 5 Formato Arquivo Texto (Novo Jogo)

O formato aceito deve ser um conjunto de 9 linhas por 9 colunas, cada um contendo um número entre 0 a 9. O valor 0 é um número especial indicando que aquela posição está vazia. Como exemplo, são fornecidos 3 jogos no formato de entrada: 01.txt, 02.txt e 03.txt.

## 6 Formato Arquivo Binário (Continuar Jogo)

O arquivo deve possuir o seguinte formato binário (considere um inteiro como sendo 4 bytes e um char como 1 byte):

- Os primeiros 4 bytes armazenam a quantidade de jogadas (é salvo um inteiro);
- Após estes 4 bytes, são armazenados os 81 bytes com o estado inicial do quadro;
- Para cada jogada que for realizada salva-se novamente ao final do arquivo o estado do quadro (81 bytes);

Você deve adotar este formato de arquivo pois o arquivo binário gerado por seu programa será testado por outros programas também.

## 7 Entrega do Trabalho

O trabalho deve ser submetido até 13 de novembro pelo AVA Moodle. O trabalho vale 10 pontos. Será descontado 2 pontos para cada dia de atraso. A entrega do trabalho consistirá em um único arquivo .zip contendo todos os códigos do seu trabalho juntamente com um breve relatório em arquivo PDF descrevendo o trabalho. O relatório deve incluir uma breve introdução, decisões de

implementação, funcionalidades não implementadas, problemas enfrentados na implementação, descrição do ambiente de desenvolvimento (sistema operacional, compilador, versão compilador), etc.

O trabalho pode ser feito individualmente ou em dupla. Casos de plágio serão tratados com rigor. Caso você faça o trabalho em grupo, submeta apenas um trabalho e identifique os componentes do grupo no relatório e no código fonte.

## 8 Fontes para Consulta

- DEITEL, P.; DEITEL, H. C: Como Programar. 6<sup>a</sup> ed. São Paulo: Pearson, 2011;
- ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C ++ e Java. 3. ed. São Paulo: Pearson, 2012;
- PIVA, D.J. et al. Algoritmos e programação de computadores. Rio de Janeiro: Elsevier, 2012.