Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação

Trabalho Prático – Jogo Sudoku
Disciplina Fundamentos de Programação (CK0211) – Semestre 2018.1

Prof. Miguel Franklin

O **Sudoku**, por vezes escrito Su Doku (数独, sūdoku), 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.

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

Na grade, algumas células já contêm números, chamados "números dados" (ou, algumas vezes, "pistas"). O objetivo é preencher as células vazias com um número em cada célula, de maneira que cada coluna, linha e região contenha os números de 1 a 9 apenas uma vez. Na solução do jogo, cada número aparece apenas uma vez em qualquer um dos sentidos (vertical ou horizontal) ou regiões. Daí vem o termo sudoku, que significa "únicos números".

Desenvolva um programa em Python (versões 2 ou 3) para o jogo Sudoku, com dois objetivos: (1) MODO INTERATIVO: Permitir que um usuário possa jogar interativamente; (2) MODO BATCH: Validar um jogo que foi dado pelo usuário.

Requisitos - MODO INTERATIVO:

1. No modo interativo, o programa deverá ser executado tendo que ler um arquivo de configuração (em modo texto) que contém a lista das pistas do jogo, podendo conter de 1 a 80 pistas, onde cada pista deve seguir o seguinte formato:

```
<COL>,<LIN>: <NÚMERO>
```

   <COL> representa a coluna da grade, em letras maiúsculas de “A” a “I”. **<LIN>** representa a linha da grade, em algarismos de 1 a 9. E <NÚMERO> representa um número de 1 a 9 dado como pista para a célula.

2. Após ler o arquivo de configuração das pistas, o programa deverá mostrar a grade do jugo, já preenchida com as pistas fornecidas no arquivo de configuração. Por exemplo, para o arquivo de configuração exemplificado abaixo:

```
A,3:5
F,1:3
D,8:7
H,6:5
```

Para o arquivo de configuração exemplificado acima, o programa deverá mostrar a grade seguinte:

```
   A   B   C    D   E   F    G   H   I
 ++---+---+---++---+---+---++---+---+---++
1||   |   |   ||   |   | 3 ||   |   |   ||1
 ++---+---+---++---+---+---++---+---+---++
2||   |   |   ||   |   |   ||   |   |   ||2
 ++---+---+---++---+---+---++---+---+---++
3|| 5 |   |   ||   |   |   ||   |   |   ||3
 ++===+===+===++===+===+===++===+===+===++
4||   |   |   ||   |   |   ||   |   |   ||4
 ++---+---+---++---+---+---++---+---+---++
5||   |   |   ||   |   |   ||   |   |   ||5
 ++---+---+---++---+---+---++---+---+---++
6||   |   |   ||   |   |   ||   | 5 |   ||6
 ++===+===+===++===+===+===++===+===+===++
7||   |   |   ||   |   |   ||   |   |   ||7
 ++---+---+---++---+---+---++---+---+---++
8||   |   |   || 7 |   |   ||   |   |   ||8
 ++---+---+---++---+---+---++---+---+---++
9||   |   |   ||   |   |   ||   |   |   ||9
 ++---+---+---++---+---+---++---+---+---++
   A   B   C    D   E   F    G   H   I
```

3. A primeira tarefa do programa após mostrar a grade com as pistas é validar se a grade está ou não de acordo com as regras do jogo. Por exemplo, se um quadrante da grade tiver algum número repetido ou se houver número repetido em uma coluna ou linha da grade, o jogo está inválido. O programa deverá verificar também se a quantidade de pistas fornecidas está de acordo com o intervalo [1;80]. Nos casos de entradas inválidas, o programa deverá informar isso ao usuário e terminar o programa.
4. Caso a grade com as pistas seja válida, o usuário deverá entrar suas jogadas, informando de uma vez só coluna, linha e número para preenchimento da grade. A entrada deverá ser semelhante ao formato do arquivo de configuração (Exemplo: A,3:7). No entanto, o programa deverá ser tolerante a diferenças de formatação. Por exemplo, o programa deve aceitar entradas como: “a, 4: 8”, “C , 2 : 8”, etc. No entanto, o programa deve criticar e não aceitar jogadas inválidas como: “K,3:8”, “A,3:12”, “C,10:8”, e solicitar a entrada de uma nova tentativa de jogada.
5. Após cada jogada com formato válido, o programa deverá verificar se a jogada fere ou não as regras do jogo. Se ferir, o programa deve mostrar mensagem de erro e solicitar uma nova jogada. Caso a jogada não fira as regras, o programa deverá mostrar a grade atualizada e pedir uma nova jogada. Uma jogada que preenche uma célula que já foi preenchida pelo usuário deverá sobrescrever o valor existente. Uma jogada em uma célula que contém uma pista deverá ser considerada uma jogada inválida.
6. O usuário poderá solicitar que uma determinada jogada prévia seja apagada, entrando o seguinte formato de comando:

```
D<COL>,<LIN>
```

No caso, o programa deverá remover o número presente na coluna **<COL>** e linha **<LIN>**. No entanto, isso só deve ocorrer se na referida célula houver um número já preenchido pelo usuário. Uma pista não poderá ser apagada.

7. O programa deverá receber jogadas em sequência até que a grade esteja completa e,

consequentemente, correta com relação às regras do jogo. Uma mensagem de sucesso deverá ser apresentada, e o programa terminará.

Requisitos - MODO BATCH:

1. No modo BATCH, o programa deverá receber dois arquivos texto, ao invés de um só. O primeiro arquivo, como no modo INTERATIVO, deverá conter a lista de pistas, no mesmo formato. O segundo arquivo deverá conter uma lista de jogadas, onde cada jogada deverá cumprir a mesma formatação da lista de pistas.
2. O programa inicialmente deverá validar o arquivo de pistas, buscando casos onde uma ou mais pistas firam a regra do jogo. Caso haja pista(s) ferindo as regras, o programa deverá exibir mensagem de erro e terminar.
3. Caso o arquivo de pistas esteja correto, o programa deverá ler todo o arquivo e jogadas e, ao final, deverá informar as jogadas que são inválidas e, ao final, se a lista de jogadas fornecidas preenche ou não a grade, concluindo o jogo. A formatação das mensagens é estritamente rígida. As mensagens deverão ser exibidas conforme os exemplos a seguir:

```
A jogada (A,3) = 5 eh invalida!
A jogada (F,4) = 2 eh invalida!
A jogada (I,8) = 1 eh invalida!
A jogada (A,3) = 5 eh invalida!
A grade foi preenchida com sucesso!
```

```
A jogada (C,3) = 2 eh invalida!
A jogada (G,1) = 7 eh invalida!
A grade foi nao foi preenchida!
```

Não utilizar qualquer tipo de acentuação ou caracteres especiais.

4. As jogadas inválidas seguem as mesmas regras do modo INTERATIVO (Requisito 5 do Modo INTERATIVO).
5. A correção da saída do programa no modo BATCH será realizada de forma automática. Portanto, é necessário observar se a formatação exemplificada no item 3 está sendo devidamente cumprida.

Critérios de Avaliação

A avaliação será realizada em duas fases:

1. Análise do código-fonte;
2. Análise da execução do programa (teste interativo e teste automático).

O código-fonte será avaliado de acordo com os seguintes critérios qualitativos:

i. Eficácia do programa em suprir todos os requisitos;
ii. Eficiência do programa (otimização);
iii. Organização do código (uso racional de subprogramas, estruturas, etc.);
iv. Legibilidade do código (uso de endentação e semântica dos identificadores de variáveis);
v. Documentação (comentários dentro do código fonte).

Obviamente, funcionalidades adicionais às que foram solicitadas neste documento são bem vindas e serão gratificadas na nota (na medida do possível). O código-fonte deve conter, em comentário no início, os nomes e matrículas dos alunos que compõem o grupo. O código-fonte deve ser submetido na data fixada através de servidor de *upload*, a ser definido.

Lembramos que todos os programas serão submetidos a análises sintática e léxica automáticas, que podem evidenciar plágio.

A versão utilizada do Python (Python 2 ou Python 3) deverá ser informada nos comentários no início do código-fonte.

Os trabalhos serão corrigidos no **Linux**. Portanto, certifique-se que o trabalho feito no Windows também roda no Linux.

EQUIPES DE ATÉ 3 (três) ALUNOS!!!

Prazo de Entrega: 22 de junho de 2018, até 23:59.

Upload através do SIGAA (Turma Virtual)

Não serão aceitas entregas por e-mail.