

Trabalho de Linguagens de Programação

Descrição

A intenção do trabalho é proporcionar ao estudante um contato inicial com o paradigma funcional, em especial a linguagem Haskell. O trabalho consiste na implementação de um dos jogos descritos abaixo. A seguir as especificações:

- Deve ser feito em dupla;
- O tema do trabalho será sorteado em sala, os alunos faltantes deverão indicar a dupla até 06/10, o sorteio do tema será feito ao final da próxima aula;
- Os trabalhos serão apresentados em sala, individualmente, nos dias **18/11 e 20/11**;
- A entrega do trabalho deve ser feita pelo moodle até as **23:55 de 17/11**. Não serão aceitas entregas posteriores;
-

Campo Minado

https://pt.wikipedia.org/wiki/Campo_minado

Descrição

O programa deve implementar um tabuleiro de 20x20 com 40 minas posicionadas aleatoriamente. Ao iniciar o programa deve imprimir o tabuleiro e as coordenadas. O programa deve então receber como entrada uma coordenada e reimprimir o tabuleiro processado de acordo com as regras do jogo. Por fim o programa deve indicar se o jogador perdeu (acertou uma mina) ou ganhou (restam apenas minas no tabuleiro). **Seguir o exemplo para impressão do tabuleiro.**

Entrada: nenhuma, o jogador deve escolher as coordenadas durante a execução do programa

Saída: “Parabéns você ganhou!” ou “Você perdeu.”

Exemplo:

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

Escolha uma coordenada (x,y):

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

Escolha uma coordenada (x,y):

Sudoku Solver

<https://pt.wikipedia.org/wiki/Sudoku>

Descrição

O programa deve receber uma matriz de 9x9 de entrada, com o estado inicial de um tabuleiro de sudoku, preencher os espaços em branco de acordo com a regra do sudoku e imprimir na tela a matriz com todos os espaços preenchidos.

Entrada: matriz de 9x9 representando cada item do tabuleiro, itens em branco são vazios

Saída: tabuleiro completo com os números da mesma linha separados por espaço

Exemplo:

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

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

Entrada:

```
[[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]]
```

Saída:

```
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
```

Reversi IA

<http://www.othellobrasil.com.br/regras.php>

Descrição:

Reversi é um jogo de dois jogadores (veja as regras no link) em um tabuleiro de 8x8, o programa deve implementar uma inteligência artificial para jogar contra um jogador. Utilize “X” para o computador e “O” para o jogador.

A cada rodada o programa deve:

1. Pedir ao jogador as coordenadas de onde colocar a peça
2. Reempirir o tabuleiro com a peça posicionada e, quando necessário, as peças viradas.
3. Pedir que o jogador aperte enter
4. Imprimir o tabuleiro após a jogada do computador, indicando as coordenadas escolhidas;

Dicas: implemente primeiramente o jogo para dois jogadores e apenas depois implemente a inteligência artificial. Para testar implemente com um tabuleiro pequeno.

Exemplos:

	0	1	2	3	4	5	6	7	8
0	0	0	0						
1							X		
2				X					
3				X					
4				X					
5				X					
6									
7									
8									

Jogada do computador: (6,1)

	0	1	2	3	4	5	6	7	8
0	0	0	0						
1									
2				X					
3				X					
4				X					
5				X					
6									
7									
8									

Escolha uma coordenada (x,y):

Binary Puzzle Solver

<http://www.binarypuzzle.com/rules.php>

Descrição:

Este jogo é parecido com sudoku, mas as células podem ter apenas 0 e 1, veja as regras no link acima. Implemente um programa que recebe como entrada a situação inicial do tabuleiro, e retorne como saída o tabuleiro totalmente preenchido. Utilize tabuleiros de 8x8.

Dica: utilize as dicas (tips) do site para diminuir a quantidade de tentativas

Entrada:

Uma matriz indicando cada posição do tabuleiro e seu valor

Saída:

O tabuleiro completamente preenchido

Exemplo:

	0						
			1		1		0
		0					
	1						
				1			
	0				1		
	0			0			
				0		0	

1	0	1	0	1	0	1	0
0	1	0	1	0	1	1	0
1	0	0	1	1	0	0	1
0	1	1	0	0	1	1	0
0	1	0	1	1	0	0	1
1	0	1	0	1	1	0	0
1	0	0	1	0	0	1	1
0	1	1	0	0	1	0	1

Entrada: [[0,,,,],[,1,1,0],[,0,,,,],[1,,,,],[0,,,,],[,1,1,0],[,0,,,,],[1,,,,]]

Saída:

```
1 0 1 0 1 0 1 0
0 1 0 1 0 1 1 0
1 0 0 1 1 0 0 1
0 1 1 0 0 1 1 0
0 1 0 1 1 0 0 1
1 0 1 0 1 1 0 0
1 0 0 1 0 0 1 1
0 1 1 0 0 1 0 1
```