MC102 - Algoritmos e Programação de Computadores

Jogo da Vida

Nesta tarefa, exercitaremos o uso estruturas multidimencionais em Python. O tema será o autômato celular proposto por John Horton Conway em 1970, conhecido como Jogo da Vida.

O jogo, que na verdade não tem jogadores, opera sobre um tabuleiro com células vivas e células mortas. Utilizaremos um padrão ASCII-ART em que células vivas são representadas por caracteres @ e células mortas por espaços em branco. Uma moldura com caratecteres + e - será utilizada para facilitar a visualização da delimitação dos diagramas.



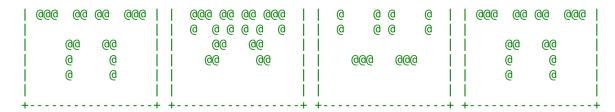
Na definição original, o diagrama é infinito e toda célula tem 8 vizinhos como esquematizado abaixo:

No nosso caso, para simplificar, consideraremos que as células adjacentes à moldura estão sempre mortas. Os estados das outras células poderão ser modificadas a partir da seguintes regras:

- Sobrevivência: toda célula viva com dois ou três vizinhos sobrevive para a próxima geração.
- Morte: toda célula com quatro ou mais vizinhos irá morrer por superpopulação; toda célula com menos de dois vizinhos irá morrer por isolamento.
- Nascimento: uma célula morta com exatamente três vizinhos irá (re)nascer.

Todas as mortes e todos os nascimentos são calculadas simultaneamente, dando origem a uma sequência de quadros.

+			+	+			+	- +					+ +					+
İ	@	@	į	İ			į	j							@	@		İ
j	@	@	Ĺ	ĺ	@@	@@	İ	Ĺ		@@@	<u>@@@</u>		ÌΪ		@	@		Ĺ
j	00	@ @	ĺ	ĺ	@ @	@ @	j	Ĺ					ÌΪ		@ @	@ @		Ĺ
				@	0 0	0 0	@		@	@	0	@						
	000 000	00 (000	@@	90 00	@@ @@	1 96		@	@	0	@		666	@ @	@ @	666	
	0 0 0	0 0 0	<u>a</u>	@	9 0 0	0 0 0)		@	@	@	@		@	0 0	0 0	@	
	60	@@		1	@@@	<u>@@@</u>				000	<u>@@@</u>				@ @	@ @		
				1														
	00	@ @			@@@	000				@@@	666		$ \cdot $		@ @	@ @		
	0 0 0	0 0 0	<u>a</u>	@	9 0 0	0 0 0)		@	@	@	@	$ \cdot $	@	0 0	0 0	@	



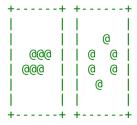
Entrada

As primeiras linhas da entrada conterão um diagrama inicial para o jogo da vida no formato utilizado nos exemplos anteriores. A última linha conterá o número de passos a serem processados na sequência, ou seja, o número de quadros além do quadro original que você deverá apresentar na saída.



Saída

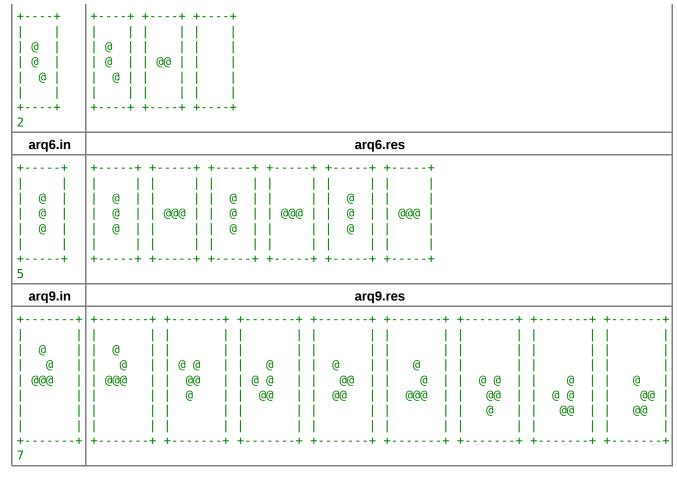
A saída conterá uma repetição do diagrama inicial seguida do(s) quadro(s) solicitado(s).



Testes para o SuSy

Esta tarefa terá 10 testes abertos com padrões conhecidos do Jogo da Vida. Note que alguns padrões estabilizam, outros desaparecem, outros oscilam e outros se deslocam no diagrama. Veja, como exemplos, os testes 2, 3, 6 e 9.

arq2.in	arq2.res										
++	++ ++										
arq3.in	arq3.res										



Esta tarefa inclui dois testes fechados.

Dicas de Python 3 para esta tarefa:

- Cada quadro pode ser armazenado como uma lista de lista de caracteres (" " ou "@") ou inteiros (0 ou 1).
- Exercite o uso de estruturas com três dimensões armazenando a sequência de quadros como uma lista de quadros.
- Ao trabalhar com listas, precisamos diferenciar as operações que geram novas listas das que criam referências distintas para a mesma lista. Observe os exemplos:

```
$ python3
>>> a = [0, 0, 0]
>>> b = a
>>> b[0] = 1
>>> print(a)
[1, 0, 0]

>>> c = [0, 0, 0]
>>> d = c.copy()
>>> d[0] = 1
>>> print(c)
[0, 0, 0]
```

• Recomenda-se a escrita de funções para estruturar o seu programa.

 O uso de outras bibliotecas padrão Python para resolver esta tarefa é permitido, mas não é necessário.

Orientações para submissão

Veja <u>aqui</u> a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar <u>main.py</u>. No link <u>Arquivos auxiliares</u> há um arquivo <u>arqs-09.zip</u> que contém todos os arquivos de testes abertos e seus respectivos resultados compactados. Os arquivos executa-testes.py e executa-testes-windows.py também estão neste pacote.

Observe o limite máximo de 20 submissões

A nota final é proporcional ao número de testes que executaram corretamente. A submissão de um código que não implementa o algoritmo solicitado, mas que exibe as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada será considerada fraude e acarretará a atribuição de nota zero à média final da disciplina.

O peso desta tarefa é 3.

O prazo final para submissão é 10/06/2018.