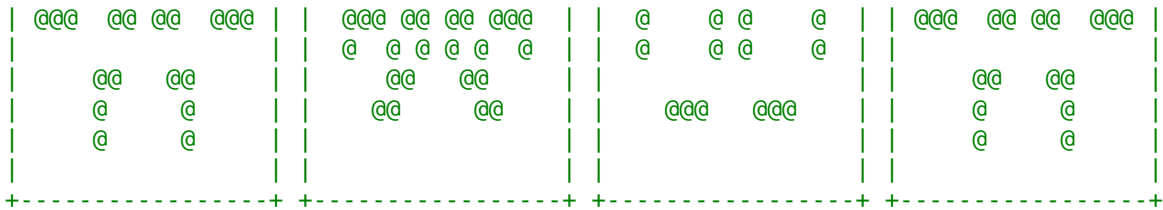


# Jogo da Vida

A diagram of a 2D hexagonal lattice. The lattice is represented by a grid of dashed lines forming hexagons. Three atoms, labeled 'a', are positioned at the center, top, and bottom-left vertices of a central hexagon.

$$\begin{array}{ccccc}
 & V_8 & & V_1 & & V_2 \\
 & & \backslash & | & / & \\
 V_7 & - & @ & - & V_3 \\
 & / & & | & \backslash & \\
 & V_6 & & V_5 & & V_4
 \end{array}$$

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|



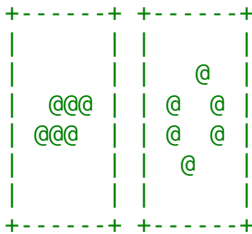
## Entrada

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 |
|----------|----------|
| <p>2</p> |          |
| arq3.in  | arq3.res |

|  |  |
|--|--|
| <pre> +---+         @     @     @         +---+ 2 </pre> | <pre> +---+ +---+ +---+                     @                 @     @           @                                 +---+ +---+ +---+ </pre>   |
| <b>arq6.in</b>   | <b>arq6.res</b>  |
| <pre> +---+         @     @     @         +---+ 5 </pre> | <pre> +---+ +---+ +---+ +---+ +---+ +---+ +---+                                       @           @           @           @     @     @     @     @     @           @   +---+ +---+ +---+ +---+ +---+ +---+ +---+ </pre>   |
| <b>arq9.in</b>   | <b>arq9.res</b>  |
| <pre> +---+         @     @     @         +---+ 7 </pre> | <pre> +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+   @       +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ </pre> |

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 [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar `main.py`. No link [Arquivos auxiliares](#) há um arquivo [args-09.zip](#) 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.

---