

Instituto de Computação - Unicamp

MC102 - Algoritmos e Programação de Computadores

Laboratório 13 - Walking Dead

Prazo de entrega: **17/05/2019 23:59:59**

Peso: **1**

Professor: Luiz Fernando Bittencourt *Professor:* Eduardo Xavier

Descrição



O apocalipse está sobre nós, e ele chegou na forma de um ataque de zumbis! Em preparação para o apocalipse zumbi, um grupo de cientistas estudou padrões de propagação e interação entre humanos e zumbis. Em um modelo simplificado, podemos representar os indivíduos do planeta em uma grade (matriz), e prever o destino de um indivíduo x após um dia baseando-se na sua vizinhança atual (veja figura abaixo).

v_1	v_2	v_3
v_4	X	v_5
v_6	v_7	v_8

Um indivíduo x que esteja numa posição (i, j) desta matriz, tem como vizinhos àqueles nas posições $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$, $(i+1, j+1)$ que correspondem respectivamente aos vizinhos v_1 , v_2 , v_3 , v_4 , v_5 , v_6 , v_7 , e v_8 da figura.

Os cientistas concluíram que as seguintes interações modelam a relação humano/zumbi:

- Se x for humano e possuir pelo menos um vizinho zumbi, então x é infectado e se torna um zumbi no dia seguinte;
- Se x for zumbi e possuir dois ou mais vizinhos humanos, ele é caçado e morto pelos humanos;
- Se x for zumbi e não possuir nenhum vizinho humano, ele morre de fome e fica vazio no dia seguinte;
- Se x estiver vazio e possuir **exatamente** dois vizinhos humanos, independente dos demais vizinhos serem zumbis ou vazio, então um humano nasce em x no dia seguinte.
- Se nenhuma das alternativas anteriores for verdade, então x permanece como está.

O estado da população em um determinado dia é baseado apenas nas informações de vizinhança do **estado da população no dia anterior**.

Veja os exemplos abaixo. Nestes exemplos H identifica um humano e Z um zumbi. Estes exemplos analisam apenas as mudanças em x de um dia para o outro. Como dito anteriormente, é preciso analisar individualmente cada posição da matriz para modificar o seu estado para o próximo estado da população.

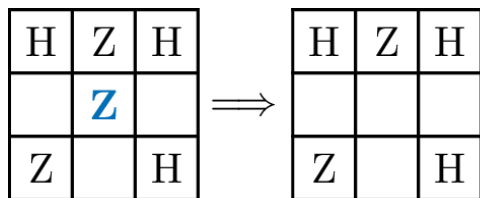
Exemplo 1: Suponha que no estado atual, x é humano, e possui ao menos um vizinho zumbi. Portanto x será um zumbi no próximo dia.

H	Z	H
	H	
Z		H

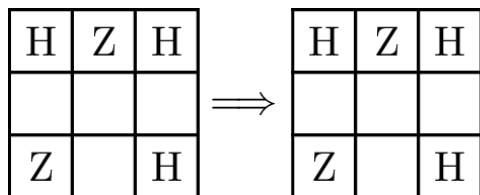
 \Rightarrow

H	Z	H
	Z	
Z		H

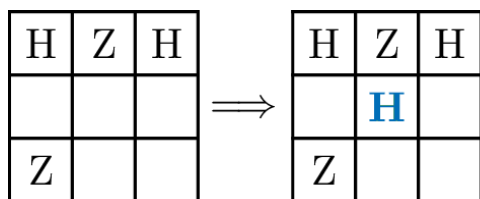
Exemplo 2: Suponha que no estado atual, x é um zumbi, e possui ao menos dois vizinhos humanos. No estado seguinte x será o vazio pois será morto.



Exemplo 3: Suponha que no estado atual, x está vazio. Como não possui exatamente dois vizinhos humanos, nada acontece.



Exemplo 4: Suponha que no estado atual, x está vazio, e possui exatamente dois vizinhos humanos. No dia seguinte nasce um humano em x .



Você deve ajudar os cientistas a preverem qual será o estado do planeta após a passagem de alguns dias. Para isso você deve criar um programa de simulação de interações humano/zumbi. O seu programa deverá ter um estado inicial do planeta carregado na forma de uma matriz, e então deverá atualizar o estado do planeta para cada dia transcorrido, onde um novo estado de um determinado dia, consiste em atualizar os valores de cada posição (i, j) considerando o estado do planeta no dia anterior.

Objetivo

O objetivo deste laboratório é, dada a configuração inicial da população, simular o estado da população durante alguns dias utilizando as regras de interação humano/zumbi definidas acima.

Entrada

A primeira linha da entrada é composta por dois inteiros, m e n , representando respectivamente o número de linhas e colunas da matriz. A segunda linha contém um inteiro i , representando o número de dias que desejamos simular o estado da

população. À seguir temos cada uma das m linhas da matriz. Cada linha é composta por n inteiros, onde 0 representa o vazio, 1 um humano e 2 um zumbi.

Note que elementos na primeira ou última linha, assim como elementos na primeira ou última coluna, não possuem todos os vizinhos. *Dica:* Para simplificar o processo de verificação deste casos, você pode utilizar uma matriz estendida com duas linhas e colunas adicionais, ambas preenchidas com zeros.

Saída

A saída deverá reportar o estado da população para cada um dos i dias transcorridos. Para cada estado à ser impresso, você deve primeiro imprimir `iteracao x`, substituindo `x` pelo número do dia, seguido pela impressão da matriz. Você deve imprimir inicialmente como `iteracao 0`, o estado inicial da população.

Exemplos

Teste 01

Entrada

```
18 26
6
0 2 2 2 1 1 1 1 0 0 0 1 2 2 1 2 0 1 1 1 1 0 1 0 1
1 1 1 2 1 2 0 1 1 1 0 0 1 2 0 1 2 1 2 0 0 1 1 1 2 2
1 1 0 1 2 0 2 2 2 0 2 2 2 1 0 0 1 0 0 1 0 0 2 1 2 0
0 2 0 2 1 2 1 1 2 2 2 0 0 0 2 1 0 2 2 2 2 0 0 1 1 1
0 2 0 2 1 1 1 0 0 2 2 2 2 0 2 2 2 0 2 0 2 0 0 1 2 0
1 0 1 1 1 2 1 0 2 2 0 1 0 2 2 2 1 0 2 1 0 0 1 2 2 0
0 1 1 2 2 2 0 2 2 2 1 2 2 0 2 2 0 1 2 2 1 0 2 2 2 2
2 1 0 0 2 0 2 1 0 2 1 2 1 1 1 2 0 2 2 0 0 0 2 2 2 0
0 0 2 1 1 2 1 2 1 0 2 2 1 1 0 2 2 0 2 1 0 1 2 2 1 2
1 2 1 1 0 1 2 1 0 0 2 0 1 0 2 1 2 0 2 0 1 0 0 1 0 0
0 0 1 1 0 1 0 1 1 0 2 1 0 0 2 2 1 2 1 2 1 2 1 1 2 0
1 0 2 2 1 0 1 0 0 1 1 1 1 2 2 2 2 1 0 2 1 2 0 1 2 0
0 2 0 0 0 2 2 1 2 2 1 0 2 0 0 0 1 1 2 0 1 2 2 0 0 0
0 1 0 1 2 1 1 2 1 0 0 2 0 0 1 2 0 0 1 2 1 0 1 0 2 2
2 2 0 2 1 0 0 2 1 2 1 0 2 0 1 1 2 0 0 2 2 0 0 1 0 2
2 1 1 2 1 2 1 0 2 0 0 2 1 1 2 0 1 1 2 0 1 2 0 1 0 1
1 0 0 1 2 2 1 1 0 0 0 1 2 0 1 2 2 2 0 0 0 2 1 1 2 1
1 2 2 0 2 1 0 0 2 2 0 2 2 2 1 1 2 2 2 0 1 0 1 1 1 2
```

Saída

iteracao 0

```
02221111000122120111110101
11121201110012012120011122
11012022202221001001002120
02021211222000210222200111
02021110022220222020200120
10111210220102221021001220
01122202221220220122102222
21002021021211120220002220
00211212102211022021012212
12110121002010212020100100
00110101102100221212121120
10221010011112222102120120
02000221221020001120122000
01012112100200120012101022
22021002121020112002200102
21121210200211201120120101
10012211000120122200021121
12202100220222112220101112
```

iteracao 1

```
10002221011200200222110202
22202002221120020200022202
22020000012202002112110201
10102022200000021222200222
00102220002220200020200201
20222021021200022102112020
02200010202000001200212200
02000102102022220201012220
10022020212022002022020022
20220202012020020100200200
10220202100201202020202200
21002020022222000200200200
10110002002001012200200100
02120220201001200020202120
00002000202100220010000202
02202020011022002220200202
20020022000200200001102202
20200201200220222000202220
```

iteracao 2

```
20000022122000000002220000
22000002002220000010000002
20020000020000000220220002
20210000200000022000010000
00210020002200001120010002
00220002022200020202220020
00000120012000002011020000
02000210212000020202020000
20002020020000002002120000
```

0100000020200200002000000000
21200002210002202020000000
020120000200000000000000200
20220000002012022000001210
00200000002102000020000200
00000000200210200020000200
00000000022120000200000000
00000022111200000002200000
00000002200000000001000000

iteracao 3

00000002220000000002000000
00000002000000000020000000
00020000000000000220220000
00021000000000020100121000
00021000000000002220121000
00221000022000021000020000
00000200120000000022000000
00000020020000000210120000
20000020020000000002220000
12100000020000000000000000
02000000220000000000000000
02122000020000000000000000
00220000002122000000002020
00000000002200000000000000
00000000001020000000000000
00000000100200000000000000
00000002222000000002200000
00000002001000000002000000

iteracao 4

00000000000000000000000000
00000000000000000000000000
00020000000000000220200000
00002100000000000201202100
00002000000000000021202100
00002100020000022000000000
00000200220000000120000000
00000000020000000221220000
21000000000000000002200000
20200000000000000000000000
00110000000000000000000000
02220000000000000000000000
00220000002220000000000000
00000000000000000000000000
00000000012000000000000000
00000000210200000000000000
00000002202000000000000000
00000000020000000000000000

iteracao 5

00000000000000000000000000
00000000000000000000000000
00000000000000000202000000

```

000022000000000000012000210
0000011000000000000002000210
000022000000000002010000000
000002000000000000200000000
000000000000000000202200000
220000000000000000000200000
210100000000000000000000000
002200000000000000000000000
020000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000001200000000000000000
000000000210000000000000000
000000000202000000000000000
000000000000000000000000000
iteracao 6
000000000000000000000000000
000000000000000000000000000
000000000000000000020000000
000020100000000000022000021
00000220000000000110000021
00002010000000000020000000
00000000000000000200000000
00000000000000000000000000
221000000000000000000000000
221200000000000000000000000
000200000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000000000000000000000000
000000002000000000000000000
000000000020000000000000000
000000000020000000000000000
000000000000000000000000000

```

Para mais exemplos, consulte os [testes abertos no Susy](#).

Observações

- O número máximo de submissões é **10**.
- O seu programa deve estar completamente contido em um único arquivo denominado `lab13.py`.
- Para a realização dos testes do SuSy, a execução do código em Python se dará da seguinte forma: (Linux e OSX) `python3 lab13.py`.
- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA.

- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.