Questão 2 (vale 2,5 pontos)

Nesta questão utilizaremos a seguinte declaração

```
#define COR_AGUA 0
```

Manipularemos nesta questão uma **imagem digital** representada por uma matriz de inteiros `img` com `nL` linhas e `nC` colunas. Suporemos que todos os pixels na borda de `img` têm a cor `COR_AGUA`. Por exemplo, abaixo temos uma imagem `img` com `nL=12` e `nC=17` em que todos os pixels na borda tem a cor `COR_AGUA`.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	**1**	0	0	0	0	0	**4**	0	0	0	0	0	0	0	0
2	0	0	**1**	**1**	0	0	0	**5**	0	0	0	0	**8**	**8**	0	0	0
3	0	0	**1**	**1**	**1**	0	**6**	0	**7**	0	**8**	**8**	**8**	**8**	**8**	0	0
4	0	0	0	**1**	**1**	0	0	0	0	0	**8**	**8**	0	**8**	**8**	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	**2**	**2**	0	0	0	0	0	0	0	0	**9**	0	0	0
7	0	0	0	**2**	**2**	0	0	0	0	0	0	0	**9**	**9**	**9**	0	0
8	0	0	0	0	**2**	**2**	**2**	0	0	0	0	**9**	**9**	0	**9**	**9**	0
9	0	0	0	**3**	0	**2**	**2**	0	0	0	0	**9**	**9**	0	**9**	**9**	0
10	0	0	0	**3**	**3**	0	0	0	0	0	**9**	**9**	**9**	0	**9**	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Diremos que um **caminho** na imagem é uma sequência de pixels ligados *horizontalmente* ou *verticalmente*. Por exemplo, na imagem acima a sequência de pixels `(1,2),(2,2),(2,3),(3,3),3,4)` é um caminho, já a sequência de pixels `(2,3),(3,4)` **não** é um caminho.

Uma **ilha** na imagem `img` é um conjunto de pixels que têm uma cor **diferente** de `COR_AGUA` **e** que estão **ligados através de um caminho**. Para ilustrar este conceito, na imagem acima pintamos todos os **pixels em uma mesma ilha com uma mesma cor** diferente de `COR_AGUA`. Por exemplo, temos que

```
Numero de pixels na ilha do pixel (1,1)   = 0  (!)
Numero de pixels na ilha do pixel (2,7)   = 1
Numero de pixels na ilha do pixel (10,3)  = 3
Numero de pixels na ilha do pixel (10,10) = 16
```

Escreva uma função **recursiva** `afundaIlha` que recebe

- uma imagem `img` com a `COR_AGUA` em todos os pixels na borda; e
- a posição `(x,y)` de um pixel em `img`.

A função deve retorna o número de pixels na ilha do pixel `(x,y)`. Além disso, a função deve pintar todos os pixels na ilha do pixel `(x,y)` de `COR_AGUA`. Por exemplo, considerando a imagem acima, a chama `afundaIlha(img,10,10)` deve retornar 16 e a nova imagem `img` é

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	**1**	0	0	0	0	0	**4**	0	0	0	0	0	0	0	0
2	0	0	**1**	**1**	0	0	0	**5**	0	0	0	0	**8**	**8**	0	0	0
3	0	0	**1**	**1**	**1**	0	**6**	0	**7**	0	**8**	**8**	**8**	**8**	**8**	0	0
4	0	0	0	**1**	**1**	0	0	0	0	0	**8**	**8**	0	**8**	**8**	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	**2**	**2**	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	**2**	**2**	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	**2**	**2**	**2**	0	0	0	0	0	0	0	0	0	0
9	0	0	0	**3**	0	**2**	**2**	0	0	0	0	0	0	0	0	0	0
10	0	0	0	**3**	**3**	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0